مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (39) العدد (5) 2017
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (39) No. (5) 2017

Evaluating the Performance of Web Cache Algorithms in Developing Information Systems in Lattakia Port

Dr. Ali Suleiman[*]
Ihab Aldibaja[**]

(Received 12 / 6 / 2017. Accepted 18 / 10 / 2017)

□ ABSTRACT □

In this paper we introduce a comparison for number of web cache algorithms applied on some information systems working in Lattakia Port General Company. We compare between current situation from the point of information retrieval speed and systems status after applying some replacement policies, we suggest best algorithm between studied ones. The study was applied on port information servers that holds database and web services, we designed and implemented an infrastructure that allowed us to record access operation to pages in addition to other useful information we used in our study. Hit Rate used to compare algorithms performance, comparisons shows that GDFS (Greedy Dual Frequency Size) gave the best result regarding the speed of presenting data to the client.

Keywords: Web Cache – information Retrieval Systems, Lattakia Port – LRU – LFU – Greedy Dual.

[*] **Associate Professor, Department of Computer and Automatic Control Engineering, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.**
[**] **Postgraduate Student, Department of Computer and Automatic Control Engineering, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.**

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (39) العدد (5) 2017
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (39) No. (5) 2017

تقييم أداء خوارزميات كاش الويب في تطوير نظم المعلومات العاملة في الشركة العامة لمرفأ اللاذقية

د.علي سليمان*
إيهاب الديباجة**

(تاريخ الإيداع 12 / 6 / 2017. قُبِل للنشر في 18/ 10 / 2017)

ملخّص

تقدم هذه الورقة البحثية مقارنة لعدد من خوارزميات كاش الويب المطبّقة على مجموعة من نظم المعلومات العاملة في الشركة العامة لمرفأ اللاذقية، حيث تتم المقارنة بين الوضع الحالي لهذه النظم من ناحية سرعة استعادة المعلومات وعرضها للمستخدم وبين وضع النظم بعد تطبيق مجموعة من سياسات الاستبدال لخوارزميات الكاش مع اقتراح الخوارزمية الأفضل، تمت الدراسة على المخدمات الموجودة في مرفأ اللاذقية والتي تقدم خدمات قواعد البيانات بالإضافة لخدمات الويب الخاص بهذه النظم حيث تم تصميم وتنفيذ البنية التحتية التي تسمح بتسجيل عمليات الوصول إلى الصفحات المختلفة بالإضافة إلى تسجيل مجموعة منوعة من المعلومات المساعدة في إجراء المقارنة، اعتمد مقياس معدل الإصابة *Hit Rate* لقياس أداء الخوارزميات، وبينت القياسات أن خوارزمية *GDFS (Greedy Dual Frequency Size)* أعطت النتيجة الأفضل من ناحية سرعة تقديم البيانات المطلوبة لمستخدمي النظم.

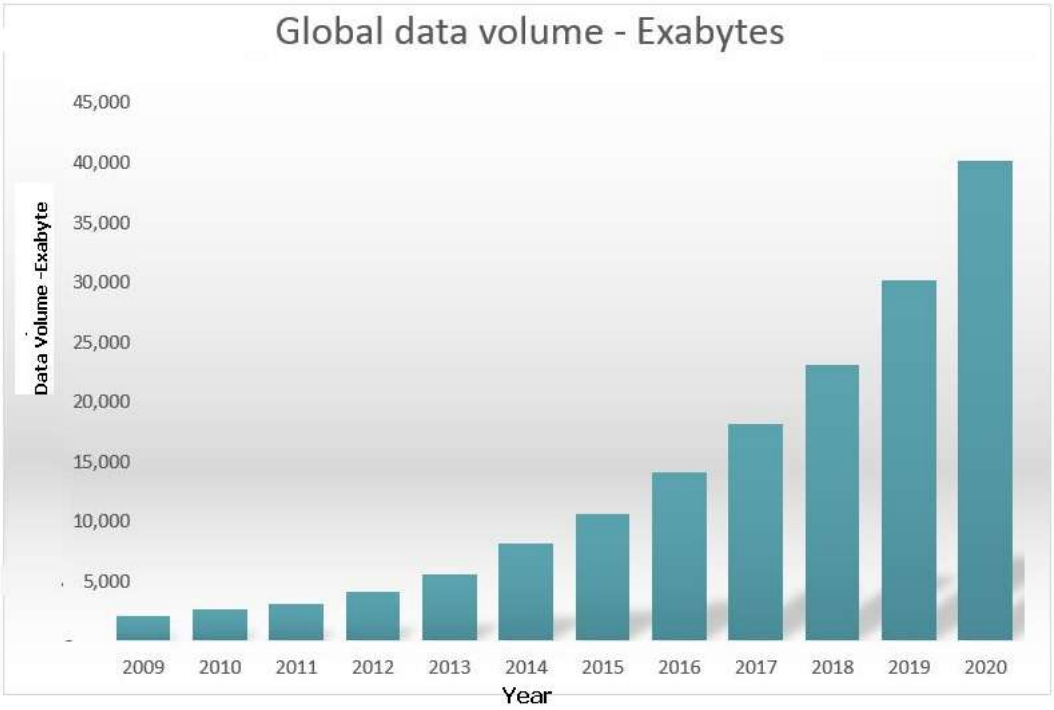
الكلمات المفتاحية:كاش الويب – نظم استعادة المعلومات – مرفأ اللاذقية –Greedy Dual–LFU–LRU.

* أستاذ مساعد – قسم هندسة الحاسبات والتحكم الآلي – كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين – اللاذقية – سورية
**طالب دراسات عليا (دكتوراه) – قسم هندسة الحاسبات والتحكم الآلي – كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين – اللاذقية – سورية

مقدمة

شهدت نظم المعلومات الإلكترونية في السنوات القليلة الماضية إقبالاً متزايداً من جانب مستخدمي المحتوى الرقمي والمتمثل في الخدمات ومصادر المعلومات الالكترونية، وبالتالي، أصبحت أداةً فعالةً للوصول إلى مصادر المعرفة وهدفاً أساسياً للبحث عن المعلومة وربط العالم ببعضه، وصاحب هذا التطور ازدياد هائل في عدد المواقع وصفحات الويب التي تعتبر واجهةَ لهذه النظم،سواءً كانت صفحات الويب موجودة على شبكة الانترنت، أو ضمن بنية أنظمة حاسوبية في المؤسسات قائمة على بيئة الويب للتفاعل مع المستخدم *web-based information system* ، كما ازداد حجم البيانات المتداولة على الشبكات والأنظمة الحاسوبية كما يبين الشكل (1)، مما تطلب وجود آليات فهرسة فعالة لتسهيل عمل محركات البحث وتسريع عملية التنقيب عن أي معلومة وسط هذا الازدحام المذهل من المعلومات، وفرض صعوبات أمام وصول المستفيدين إلى مصادر المعلومات المتاحة في الأنظمة المعلوماتية المختلفة.

إنّ إتاحة مصادر المعلومات في متناول المستفيدين، يتطلب الاستعانة بمجموعة من الأدوات والتقنيات التي من شأنها تنظيم وإدارة المحتوى الالكتروني لتحسين فاعلية عملية استعادة البيانات منها، وعلى الرغم من أنّ الإمكانيات المادية للمخدمات *Hardware Specification* آخذة في الارتفاع، إلا أنّ العديد من هذه المخدمات لم تعد قادرة على التعامل بالشكل الأمثل مع الأعداد الضخمة من طلبات البيانات التي يولدها المستخدمون بشكل متزايد.



الشكل (1) أحجام البيانات المتداولة والمتوقع تداولها في أنظمة المعلومات بين عامي2009- 2020 حسب إحصائية لشركة *Cisco* [1]

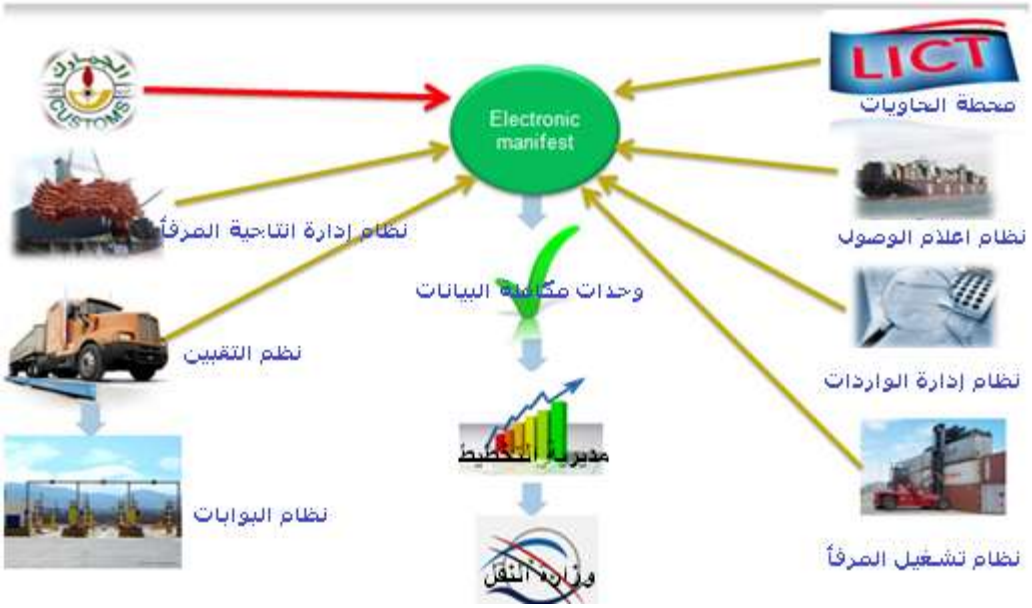
مشكلة البحث

في عام 2013 قامت الشركة العامة لمرفأ اللاذقية بالربط الشبكي مع مديرية الجمارك في دمشق للحصول على بيانات المانيفست الالكتروني *Electronic Manifest* الذي يعتبر الوثيقة الأساسية في إنجاز كافة العمليات الخاصة

بالعمل المرفئي، يضم المانيفست الالكتروني آلاف المفردات عن الحاويات *Containers* ، البضاعة *General Cargo*، عددها ووزنها ونوعها، مصدرها، وجهتها، وغير ذلك من البيانات[2].

قامت شركة المرفأ ببناء نظام شبكي على شكل تطبيق ويب يرتبط مع عدد من النظم الأخرى مثل نظام العمليات في محطة حاويات اللاذقية الدولية *LICT*، نظام الأبواب والقبابين، وغيرها كما هو مبين في الشكل (2)، كلّ هذه النظم تقوم بطلب معلومات من المانيفست الالكتروني وإضافة معلومات تشغيلية عليه، ومن ثم يتم معالجة هذه المعلومات لتقديم تقارير يومية مفصلة إلى جهات مختلفة داخل وخارج الشركة.

لاحظنا من خلال مراقبة أداء هذه النظم حصول بعض الاختناقات في الأداء نتيجة الضّغط الشديد على معلومات المانيفست، وبالتالي، كان لابد من اتخاذ إجراءات لتسريع العمل مثل تحسين الاستعلامات *Query Optimization*، وكذلك يعتبر الكاش أحد الوسائل المهمة لتحسين الأداء والحصول على المعلومات من مصادر غير مخدّم البيانات الأساسي.



الشكل (2) مخطط النظم المعلوماتية العاملة في شركة المرفأ

أهمية البحث وأهدافه:

أهمية البحث

تكمن أهميّة البحث في تقديم كاش الويب كتقنية تسمح بتخفيف الضغط على مخدّمات نظم المعلومات في بيئة عمل تتطلب عمليات مكثفة وسريعة للوصول إلى طيف متنوع وواسع من المعلومات كالمرفأ، خصوصاً في أوقات ذروة العمل، حيث يتم معالجة آلاف المعلومات المتعلقة بالفواتير وحركات إدخال وإخراج البضاعة والشاحنات بشكل لحظي، كما يحدد البحث الخوارزمية الأفضل التي تتوافق مع طريقة طلب المستخدمين للبيانات من ناحية تسلسل طلبها من المخدم وحجم المعلومات المطلوبة.

أهداف البحث

يهدف البحث إلى تقييم أداء مجموعة من سياسات الاستبدال المختلفة لخوارزميات كاش الويب بتطبيقها ضمن الأنظمة المعلوماتية العاملة في المرفأ، من خلال مقارنة الوضع الحالي لهذه النظم قبل وبعد تطبيق الخوارزمية، وتحديد الأفضل منها،والتي ستحسن وضع الخدمات التي تقدمها نظم المعلومات، وتسريع للعمل على أرض الواقع بما يتوافق مع نموذج طلبات ووصول المستخدمين *user request and access patterns*إلى هذه النظم، كما نشرح البنى التي تم تصميمها لتطبيق نظام الكاش والحصول على البيانات التحليلية التي تتضمن مفاتيح قياس أداء الخوارزميات المختلفة.

طرائق البحث ومواده

تمّ إدخال البيانات الخاصة بطلبات المستخدمين الواصلة إلى نظم المعلومات المدروسة إلى برمجية *web log expert* الخاصة بتحليل سجلات الوصول، سمحت دراسة السجلات بالحصول على مؤشرات مختلفة عن نموذج طلب المستخدم للصفحات مثل أسماء صفحات النظام المطلوبة بشكل أكبر، وعرض الحزمة المستخدمة، ونوعية البيانات المنقولة عبر الشبكة، وتمّ الاستفادة من هذه المعلومات في تحديد عينات البحث والتركيز عليها لاستخلاص النتائج وذلك بعد تصميم وبناء النظام البرمجي الذي سمح بتطبيق خوارزميات كاش الويب في النظم المدروسة.

مجتمع وعينة البحث

البحث شمل دراسة ما يقارب *500* ألف سجل خاص بطلبات ووصول المستخدمين إلى الأنظمة المعلوماتية في الشركة، والكود التالي يبين عينة من السجلات المدروسة والتي تقدمها خدمة مايكروسوفت للإنترنت *Microsoft Internet Information System (IIS)*المسئولة عن إدارة خدمات الويب على مخدمات المعلومات في الشركة، تتضمن هذه السجلات اسم الصفحة وحجمها وزمن الوصول وحجم البيانات المنقولة من مخدم الويب وطيف واسع آخر من المعلومات التي تمّ الاستفادة منها في الدراسة.

```
#Software: Microsoft Internet Information Services 7.0
#Version: 1.0
#Date: 2016-06-20
#Fields: date time s-ipcs-method cs-uri-stem cs-uri-query s-port cs-username c-ipcs(User-Agent) sc-status sc-substatus sc-win32-status time-taken
173.131.0.122 08:01:20 30-05-2016GET /admin/data_input.aspx - 903 admin1 173.131.4.23 Mozilla/5.0+(Windows+NT+6.1;+WOW64;+rv:51.0)+Gecko/20100101+Firefox/51.0 300 0 0 710
173.131.0.122 08:01:32 30-05-2016GET /admin/create_daily.aspx - 903 admin1 173.131.4.23 Mozilla/5.0+(Windows+NT+6.1;+WOW64;+rv:51.0)+Gecko/20100101+Firefox/51.0 300 0 0 250
```

الدراسات السابقة

إنّ جميع الدراسات المرجعية التي تم الاطلاع عليها ومنها الدراسات *[6][5][4][3]* والمتعلقة بكاش الويب تؤكد أهمية الكاش في تسريع الحصول على المعلومات من النظم القائمة على الويب، وتبين أهمية استخدام سياسة الاستبدال المناسبة للنظام والتي تتناسب مع حجم البيانات ونوعها وطريقة الطلب عليها، وهكذا، إذا تم الاختيار المناسب للبيانات التي يتم تخزينها في الكاش، فإنّه من الممكن الحصول على سرعة كبيرة جداً عند الوصول للبيانات.

الإطار النظري للبحث

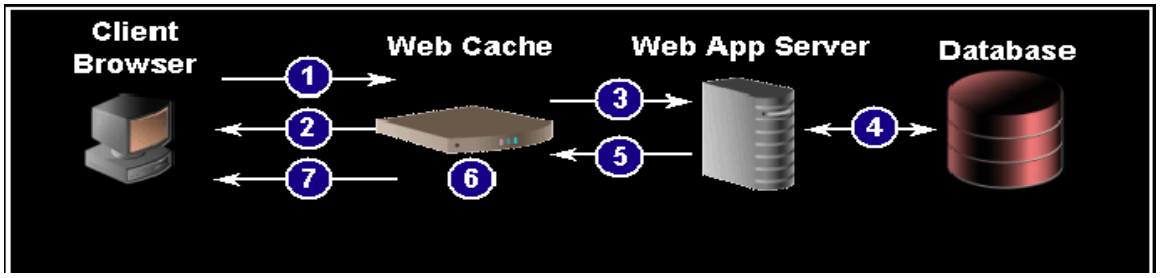
كاش الويب:

يعرّف كاش الويب أو التخزين المؤقت لصفحات وأغراض الويب بأنّه نظام حاسوبي مخصص لمراقبة طلبات الأغراض (أغراض الويب) وتخزين هذه الأغراض بحيث يتم استردادها من المخدم، حيث سيقوم الكاش بتسليم الأغراض من المخزن بدلاً من إعادة تمرير الطلب إلى المخدم المعني في الطلبات اللاحقة[7].

هنالك العديد من فوائد تطبيق خوارزميات كاش الويب، ذلك أنّ طلب البيانات وتخديمها من قبل الكاش دون العودة إلى الموقع الأصلي للبيانات يوزع الحمل ويخففه عن المصادر الأساسية التي هي عادة مخدمات تتعرض لضغط من ناحية محرك قواعد البيانات أو عرض الحزمة المستخدمة أو الموارد الأخرى كأقراص التخزين والذواكر، حيث يمكن استخدام مخازن كاش متزامنة *concurrent cache stores* وبالتالي تسريع زمن الاستجابة لطلب المستخدم بالإضافة إلى إمكانية تصميم الكاش بطريقة تستفيد فيها من بنية الشبكة لتقليل عرض الحزمة المستخدم على كامل الشبكة *[8]*.

يبين الشكل (5) نموذج مخدم – زبون مع كاش ويب حيث تتم عملية الطلب والإجابة من المخدم وفق الخطوات التالية[9]:

- يقوم الزبون بإرسال طلبات *HTTP*.
- يستجيب مخدم كاش الويب مباشرةً في حال كان الغرض متاحاً في الكاش.
- يتم تحديث البيانات التي تصف العناصر المخزنة في مخدم الكاش
- إذا لم يكن الغرض في الكاش يقوم كاش الويب بطلب الغرض من مخدم التطبيقات.
- يقوم مخدم التطبيقات بتنفيذ المطلوب والاستجابة لكاش الويب.
- إذا أمكن تخزين الاستجابة في الكاش فإنّه يتم الاحتفاظ بها من أجل طلبات لاحقة، ويتم إخراج العناصر من الكاش وفق سياسة الاستبدال للخوارزمية المحددة لعمل المخدم.
- يقوم كاش الويب بتحضير الصفحات من أجل الاستجابة لطلبات الزبائن وفق الخوارزمية المحددة له.



الشكل (5) نموذج مخدم – زبون مع تقنية كاش الويب

الخوارزميات المستخدمة في النظام

خوارزمية LRU [10]

LRU هي اختصار لـ *Least Recently Used* والتي تعني الأقل استخداماً، وقد تمّ اشتقاقها مباشرةً من التخزين المؤقت لنظام الملفات *File System Caching*. تقوم خوارزمية *LRU* على مبدأ أن الصفحات الأقل

استخداما سوف يتم إخراجها من ذاكرة الكاش، حيث أنّ الصفحات التي تم استخدامها بشكل كبير في الفترة الأخيرة يكون احتمال إعادة استخدامها مجدداً كبير أيضاً في المرات المقبلة.

Pseudo-code for LRU

If p is in the buffer then LAST(p) = current time;
Else
i) Min = current time + 1;
ii) For all pages q in the buffer do
a) If (LAST(q) < min)
victim = q
Min = LAST(q)
iii) If victim is dirty then flush it to disk
iv) Fetch p into the buffer frame held by victim
LAST(p) = current time

-خوارزمية LFU [11]

LFU هي اختصار لـ *Least Frequently Used*والتي تعني الأقل تكراراً، وهي تقوم على مبدأ استبعاد الصفحات التي تملك أقلّ عدد مرات استخدام، بغض النظر عن حداثة استخدامها مقارنة مع الخوارزمية السابقة، إذ يتم في هذه الخوارزمية الاحتفاظ بعداد من أجل كل غرض (صفحة) ويتم طرد واستبعاد الأغراض التي تملك العداد ذو القيمة الأدنى وهذه العملية تتم عند اتخاذ القرار بإجراء عملية التبديل.

LFU Pseudo-Code
Devide every value(Page) in cache by 2 if Page in cache value(Page) = R + value(Page) EXIT while cachefree<filesize(Page) find and remove candidate in cache with smallest value cachefree = cachefree + filesize(candidate) insert value(FILENAME) = R

-خوارزمية Greedy Dual [12]

إنّ خوارزمية *GreedyDual* هي تعميم للخوارزمية المعروفة *LRU* مراعيةً احتياجات التخزين المؤقت لأغراض الويب.

تقوم الخوارزمية على مبدأ الاحتفاظ بقيمة تقديرية *H(p)* لكل مستند أو صفحة *p* تم تخزينها.عندما يتم تخزين المستند تسند لقيمته التكلفة التي حصلت نتيجة تخزين *p* وبالتحديد *c(p)*. وعندما تبرز الحاجة إلى إزالة مستند من التخزين فسيتم اختيار المستند ذو قيمة *H* الأدنى، ويتم طرد هذا المستند ويتم إنقاص قيمة *H* لكل مستند متبقي في التخزين بمقدار قيمة *H* للمستند المطرود.

في أي وقت يتم فيه طلب مستند *p* تم تخزينه فإن قيمة *H(p)* تعاد إلى *c(p)*.

-خوارزمية GreedyDual-Size [13]

وهي خوارزمية معدلة عن خوارزمية *GreedyDual* ويكمن الفرق الجوهري بينهما في طريقة تركيب القيمة التقريبية *H*.

عندما نقوم بتخزين أو إعادة الوصول إلى مستند *p* ، فإن قيمة *H(p)* تصبح

$$\frac{C(p)}{S(p)} \quad (1)$$

حيث أن *S(p)* هو حجم المستند *p* وبالتالي يتم هنا أخذ حجم المستند بعين الاعتبار.

إنّ تأثير هذا التغيير في الخوارزمية الأصلية يظهر عند القيام بتخزين المستندات ذات الأحجام المتكافئة، حيث أن الكبيرة منها يرغب بإزالتها قبل الصغيرة منها (طبعاً بعد الأخذ بعين الاعتبار النفاذ إلى هذه المستندات وقيمة *H*) مما يزيد من كفاءة استخدام الذاكرة.

Algorithm GDS

Initialize L=0
Process each requested document in turn:
Let p be the current document:
IF p is already in cache
K(p) = L + C(p)/S(p)
ELSE
WHILE there is not enough space for p
Let L = mink(qj), for all qj in cache
Evict q such that K(q) = L
END WHILE
Doposit p into cache and set
K(p) = L + C(p)/S(p)
END IF

- خوارزمية GDSF[14]

GDSF هي تطوير لخوارزمية *GreedyDual* السابقة من خلال إضافة تردد طلب المستند*Frequency* إلى قيم المفتاح *Key*.

إنّ قيمة المفتاح للمستند *P* تحسب وفق المعادلة التالية :

$$K(p) = L + F(p) * C(p)/S(p) \quad (2)$$

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 لـ 1.

إذا تم طلب النفاذ إلى *p* مجدداً في الذاكرة فإن قيمة التردد الخاص به تزداد بـ *1*:

$$F(p) = F(p) + 1 \quad (3)$$

حيث أن *S(p)* هو حجم المستند *p* وبالتالي يتم هنا أخذ حجم المستند بعين الاعتبار.

البنية المصمّمة لنظام الكاش المقترح

تمّ تصميم البنية اللازمة لتطبيق خوارزميات الكاش ودمجها ضمن النظم محل البحث في مرفأ اللاذقية وذلك بعد تسجيل مؤشرات عمل هذه الأنظمة قبل تطبيق الخوارزميات، وتتكون هذه البنية من الوحدات التالية:

وحدة تخزين أجزاء الصفحات *Page Segment Storage Unit*: هي عبارة عن جدول يقوم بتخزين أقسام الصفحات التي تم طلبها من قبل الزبون والتي هي مجموعة محارف تمثل كود *HTML* للصفحة بصيغته النهائية بعد توليدها من مخدم الويب.

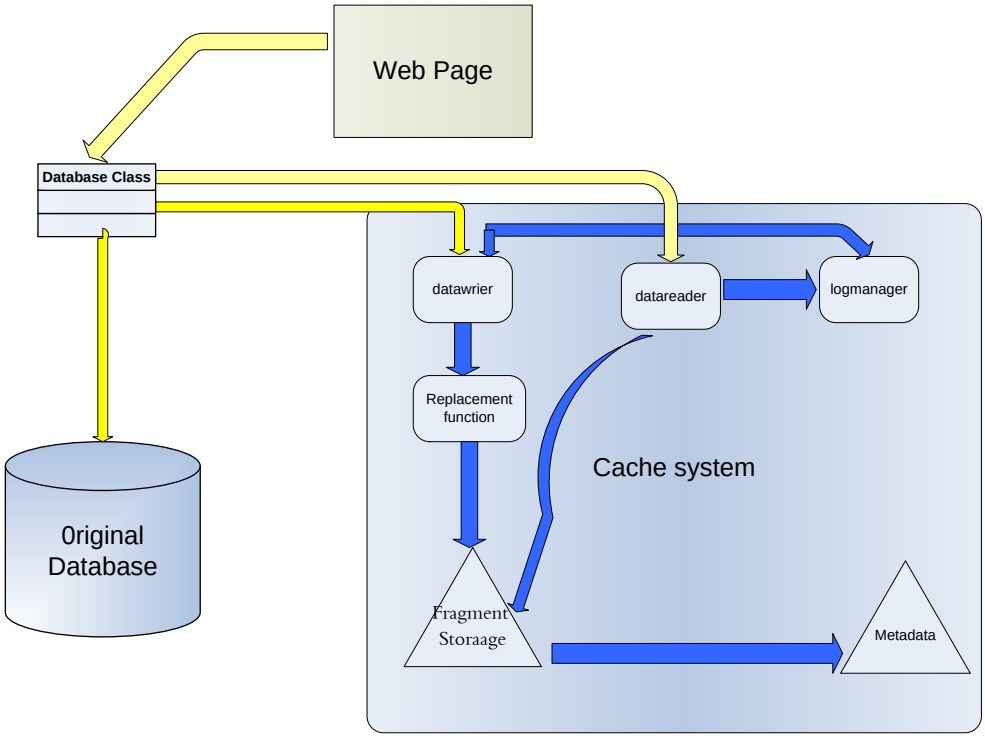
مخزن البيانات الوصفية *Data Descriptor Storage* : وهو عبارة عن قاعدة بيانات تحتوي على البيانات الفوقية *Metadata* المقترنة بجزء ديناميكي من الصفحة. وهذه البيانات تحتوي على معلومات مثل تاريخ آخر تحديث للجزء أو عدد الزيارات *hit*. إنّ هذه المعلومات مهمة للغاية من أجل سياسة الاستبدال المعتمدة لخوارزمية كاش الويب.

قارئ الكاش *Cache Reader*: يقوم هذا المكون بقراءة جزء الصفحة من الذاكرة المؤقتة ويقوم بتحديث البيانات الوصفية المرتبطة مثل عدد مرات الزيارة *Hit*.

كاتب الكاش *Cache Writer*: يقوم هذا المكون بتخزين جزء الصفحة في ذاكرة الكاش وتحديث المعلومات الوصفية المرتبطة بها (على سبيل المثال إذا كان الجزء غير موجود في ذاكرة الكاش من قبل يقوم الكاتب بإنشاء البيانات الفوقية المرتبطة بهذا الجزء، بينما إذا كان الجزء موجود مسبقاً، يقوم الكاتب هنا بعملية تعديل هذه المعلومات الوصفية).

مخزن سجل المناقلات *Log Manager Storage*: يقوم هذا المكون بتخزين الحركات التي تتم على الكاش من عمليات القراءة والكتابة والاستبعاد وهي تستخدم لأغراض الدراسة العلمية وإجراء المقارنات بين أداء السياسات المختلفة لعمل خوارزمية الكاش.

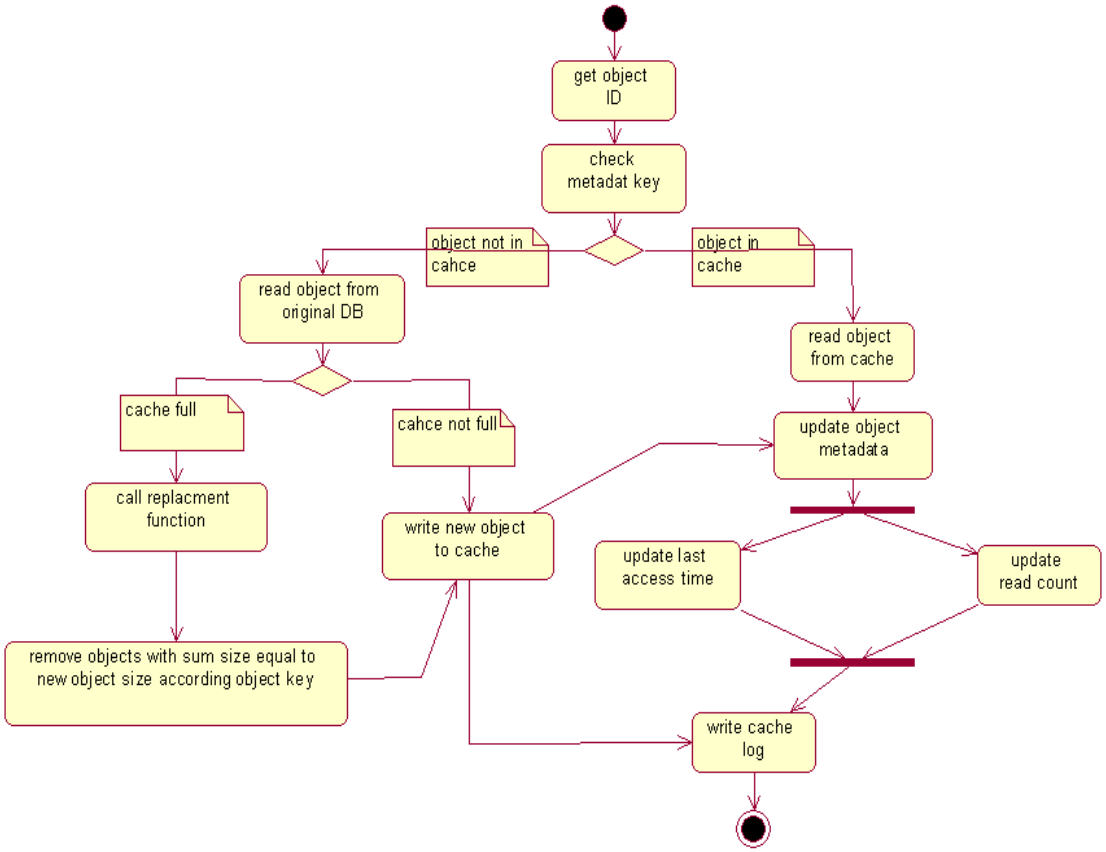
يبين الشكل (6) المخطط الصندوقي للنظام الذي قمنا بتصميمه بناءً على ما ذكرناه في فقرات البحث، حيث يظهر دور كل من مكونات نظام الكاش وكيفية تفاعلها مع بعضها، إذ يبدأ الطلب من صفحة الويب التي تتواصل مع الصفوف الخاصة بالتعامل مع قاعدة البيانات، تتولى هذه الصفوف مهمة تشغيل قارئ الكاش وكاتب الكاش بناءً على توافر البيانات المطلوبة في الكاش، يعمل كاتب الكاش وفق سياسة الاستبدال المحددة، أما سجل المناقلات *log manager* فيقوم بتسجيل المعلومات الخاصة بالوصول إلى الكاش من ناحية الزمن وحجم البيانات المتبادلة مع مخزن أجزاء الصفحات.



الشكل(6) المخطط الصندوقي للنظام

يبين الشكل (7) مخطط الفعالية لطلب الصفحة في نظام الكاش الذي تمّ تنفيذه، حيث يتم أولاً محاولة الوصول إلى البيانات من الكاش، في حال وجود الغرض في الكاش، يتم قراءة الغرض من الكاش مع تحديث المعلومات الفوقية للغرض مثل تاريخ آخر وصول وعدد مرات الوصول، في حال عدم وجود الغرض في الكاش، سوف يتم الحصول

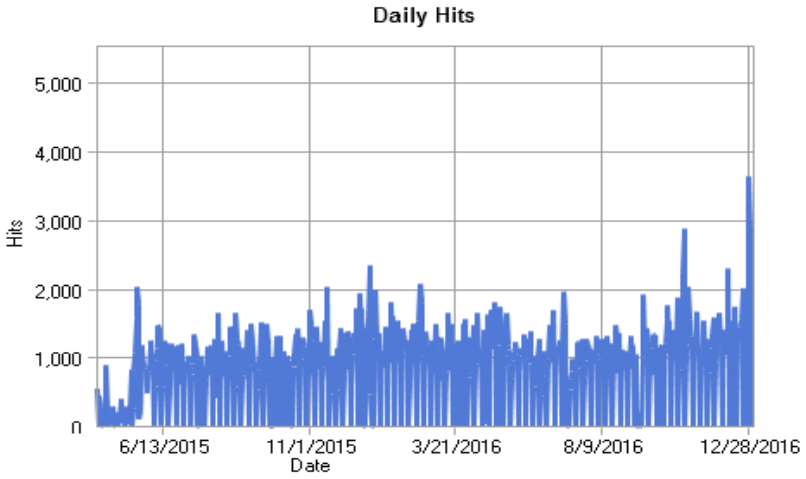
على الغرض من قاعدة البيانات الأساسية ومن ثم استدعاء تابع الاستبدال الخاص بالخوارزمية الذي يفحص إمكانية تخزين الغرض الجديد في الكاش مباشرةً أو القيام بإخراج غرض آخر أو مجموعة أغراض من الكاش في حال امتلاء الكاش وذلك وفق سياسة الاستبدال الخاصة بالخوارزمية المطبقة بحيث يكون مجموع حجوم هذه الأغراض يسمح بتخزين الغرض الجديد ضمن ذاكرة الكاش.



الشكل (7) مخطط الفعالية لطلب صفحة من نظام الكاش

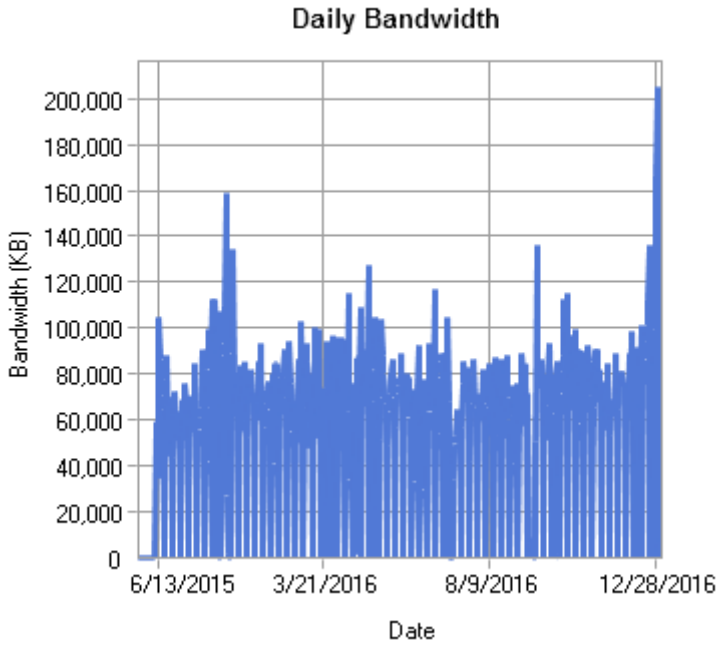
النتائج والمناقشة

تمّ دراسة البيانات الخاصة بوصول المستخدمين إلى المخدمات من أجل التعرف على طبيعة البيانات وحركتها ويبين الشكل (8) عدد مرات الوصول اليومية حسب التاريخ:



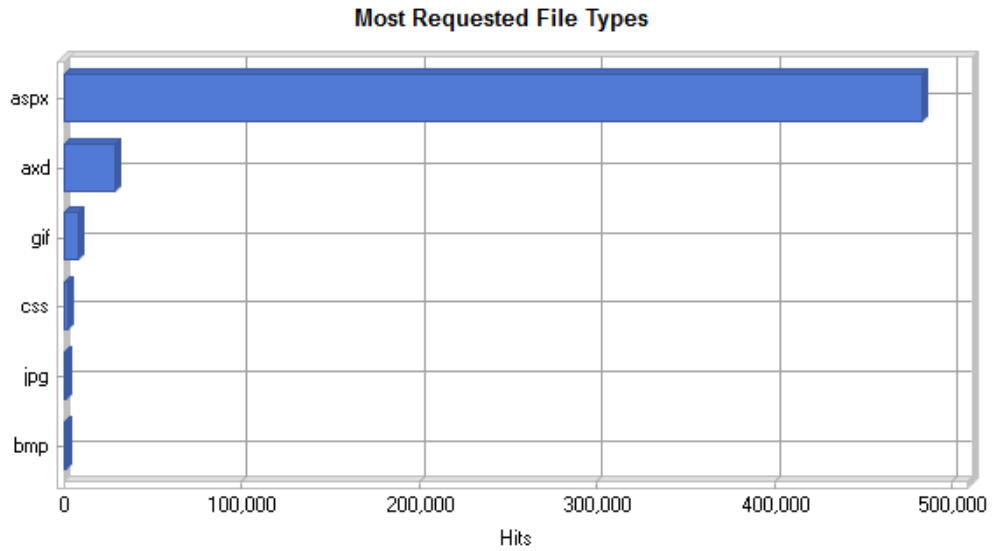
الشكل (8) عدد مرات طلبات المستخدم اليومية لصفحات النظم المدروسة حسب التاريخ

كما تمّ دراسة عرض الحزمة المستخدمة بشكل يومي كما هو مبين في الشكل (9):



الشكل (9) عرض الحزمة المستخدم من قبل المستخدمين في المرفأ حسب التاريخ

بالإضافة إلى نوعية الملفات التي يتم الوصول إليها بشكل أكثر تواتراً كما هو مبين في الشكل (10):



الشكل (10) نوعية الملفات التي تم طلبها من قبل المستخدمين حسب عدد مرات الطلب لكل نوع

وتمّ تحديد الصفحات الأكثر طلباً من قبل المستخدم لتطبيق الدراسة عليها وتدقيق النتائج التي تمّ تلخيصها في الجدول رقم (1).

الجدول (1) ملخص النتائج التي تم الحصول عليها من مراقبة النظم المعلوماتية العاملة في مرفأ اللاذقية بما يخص عمليات الوصول إلى الصفحات الخاصة بهذه النظم

Hits	
Total Hits	525,855
Average Hits per Day	625
Average Hits per Visitor	100.51
Failed Requests	2,951
Page Views	
Total Page Views	483,516
Average Page Views per Day	574
Average Page Views per Visitor	92.42
Visitors	
Total Visitors	5,232
Bandwidth	
Total Bandwidth	24.15 GB
Average Bandwidth per Day	29.41 MB
Average Bandwidth per Hit	48.16 KB
Average Bandwidth per Visitor	4.73 MB

ومنها تمّ التوصل إلى النتائج التالية بما يخص سلوك المستخدمين على الأنظمة المدروسة في المرفأ:

- معظم عمليات الوصول تتم إلى ملفات من النوع ASP والتي هي ملفات نصية، وبالتالي ركزت دراستنا على خوارزميات الكاش بما يخص تخزين الكود النصي للصفحات ولم تتطرق إلى الصور والأنواع الأخرى من الملفات.
- يعتبر عرض الحزمة bandwidth المستخدم بشكل يومي صغيراً بالنسبة لتجهيزات المرفأ الشبكية المصممة للعمل من جهة الزبون على سرعة 100 ميغابت ومن جهة المخدمات على سرعة 1000 ميغا بت، وبالتالي، فان التأخير في عرض النتائج للمستخدمين ناتج عن الحمل على قاعدة البيانات وليس عن الزمن اللازم لنقل المعلومة خلال الشبكة،وأكد هذه النتيجة دراسة عرض الحزمة المستخدم لكل طلبية للمستخدم وهي 49 كيلو بايت.
- تم تحديد الصفحات الأكثر طلباً من أجل تطبيق الدراسة عليها.
- معدل عدد مرات العرض لكل صفحة في اليوم عال وهو 574، وكل منها تتطلب الحصول على البيانات من جداول متعددة ضمن قواعد بيانات متعددة، وهذا يؤكد ضرورة تطبيق نظام كاش يقوم بمعالجة الصفحات ذات الطلب المتكرر.

تمّ تطبيق خوارزميات الكاش وفق توابع الاستبدال للخوارزميات المختلفة المذكورة على نظام المانيفست الالكتروني، يعتبر المانيفست هو الوثيقة الأساسية لتشغيل السفينة وتضم معلومات عن رحلة السفينة والبضاعة والحاويات الواصلة إلى المرفأ وتعتمد عمليات التحميل والتفريغ وإخراج البضاعة وتعيين القوى العاملة والآليات على معلومات المانيفست، وبالتالي، يعتبر من الأنظمة التي تتطلب تنفيذ استعلامات متعددة ومعقدة فضلاً عن العمليات الإحصائية المكثفة التي تتم على البيانات الواردة فيه.

تصل بيانات المانيفست إلى شركة مرفأ اللاذقية الكترونياً من المديرية العامة للجمارك في دمشق من خلال ربط الكتروني عبر شبكة ذكية، ويتم تحميل البيانات على قاعدة بيانات محلية في المرفأ مرتبطة بالأنظمة التشغيلية المختلفة (القبابين – الأبواب – الخزن – ...)، ويبلغ عدد سجلات البيانات المحملة على هذا النظام بحدود 40 مليون سجل.

تمّ اعتماد معدل Hit Rate (معدل الإصابة) والذي نحصل عليه بتقسيم عدد المرات التي تم الحصول فيها على البيانات من الكاش على عدد مرات الوصول الكلية إلى البيانات، وذلك لتقييم أداء الخوارزمية[15]، وعندما يكون مرتفعاً، فإنّ ذلك يدلّ على أنّنا استطعنا استرجاع نسبة عالية من الطلبات من التخزين المؤقّت، وهو عادةً النتيجة المرجوّة لمعظم مديري النظم، بالإضافة إلى أننا قمنا بأخذ الزمن الإجمالي لتحميل بيانات الطلب بعين الاعتبار. الدراسة تمت بعد تقسيم محتويات الصفحات المدروسة إلى وحدات منطقية تمثل كل منها وحدة معلومات جزئية يمكن أن تطلب من المستخدم أو الأنظمة الفرعية المرتبطة بالنظام المدروس بشكل منفصل، فعلى سبيل المثال، تمّ تقسيم مستند المانيفست الالكتروني الذي كان يعامل كوحدة كاملة، إلى وحدة المعلومات العامة للرحلة General visit info، وحدة البوالصMarine bill of lading unit، وحدة الحاويات Container unit، وحدة معالجة البضاعة Cargo delivery unit،...، يمكن للمستخدم أو لنظام فرعي أن يطلب غرضاً يمثل حاوية أو بوليصة، ويمكن أن يطلب صفحة المانيفست كاملاً بكل ما تتضمنه من أغراض. كلما كانت هذه الوحدات مطابقة لمتطلبات العمل ومقسمة بشكل جيد، فإن ذلك سوف يسمح للمستخدم والنظم الفرعية بالاستفادة بشكل أفضل من الأغراض المخزنة في الكاش.

أمّا بالنسبة لخوارزميات *Greedy Dual* فيما يتعلق بتابع الكلفة *H(p)*، تمّ اعتماد كلفة المعالجة *CPU Cost* بالإضافة إلى كلفة عمليات التخزين لاستعادة المعلومات *I/O Cost* والتي يقدمها محرك قواعد البيانات كمعلومات

إحصائية لدى تنفيذ استعلام كما يبين الشكل (11)، وتمّ تحميلها ضمن نظام الكاش، بحيث أصبحت قيمة المفتاح للغرض o هي:

$$K(o) = L + F(o) * (CPU\ Cost + IO\ Cost)/S(o) \quad (4)$$

وبذلك تكون قيمة المفتاح لكامل المستند p هي مجموع قيم المفاتيح للأغراض التي يتكون منها المستند:

$$K(p) = \sum_{i=1}^{n} K(o_i) \quad (5)$$

حيث n هو عدد الأغراض التي يتكون منها المستند p.

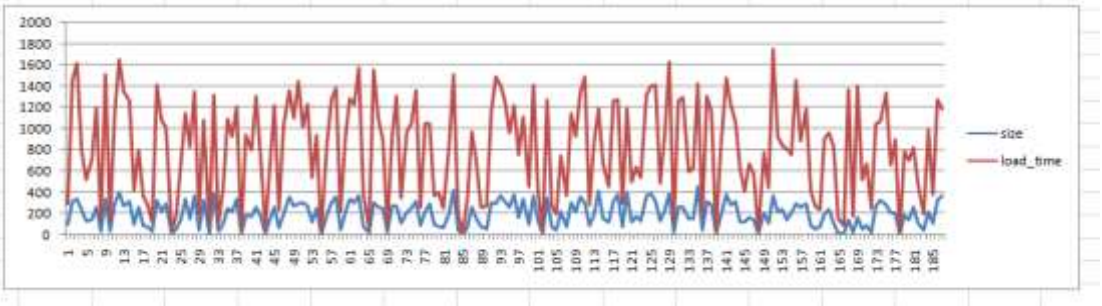
Sort

Sort the input.

Physical Operation	Sort
Logical Operation	Distinct Sort
Estimated Execution Mode	Row
Estimated I/O Cost	0.0056306
Estimated Operator Cost	0.04114 (0%)
Estimated Subtree Cost	10.0013
Estimated CPU Cost	0.0355264
Estimated Number of Executions	1
Estimated Number of Rows	3822.02
Estimated Row Size	109 B
Node ID	1

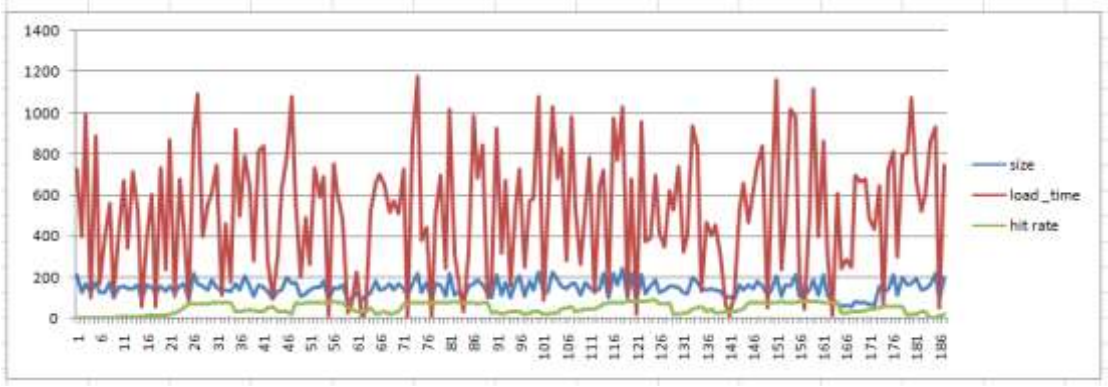
الشكل (11) مجموعة المعلومات التي يقدمها محرك قواعد البيانات SQL Server عن كلفة استعلام تم تنفيذه.

الشكل (12) يبين لنا العلاقة بين حجم المستند وزمن تحميل الصفحة بدون تطبيق نظام الكاش



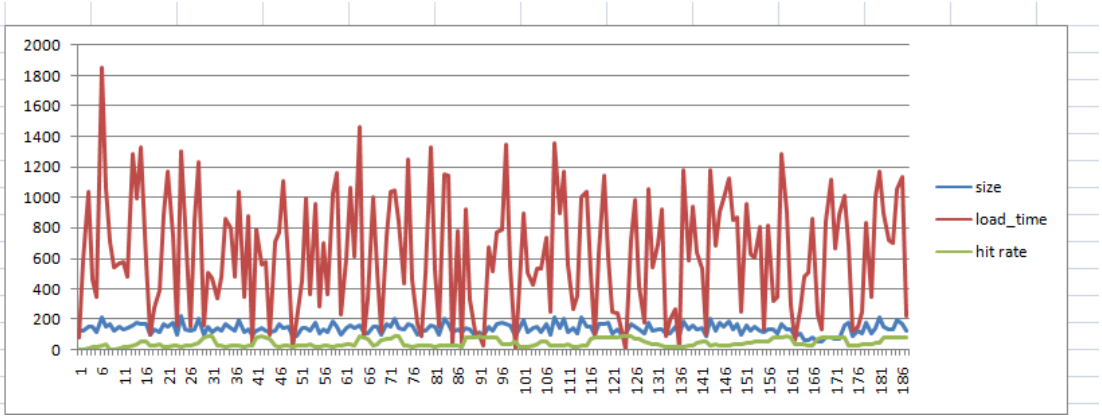
الشكل (12) حجم المستند وزمن تحميل الصفحة بدون تطبيق أي نظام للكاش

الشكل (13) يبين لنا العلاقة بين حجم المستند وزمن تحميل الصفحة ومعدل الإصابة لدى تطبيق خوارزمية LRU لنظام الكاش.



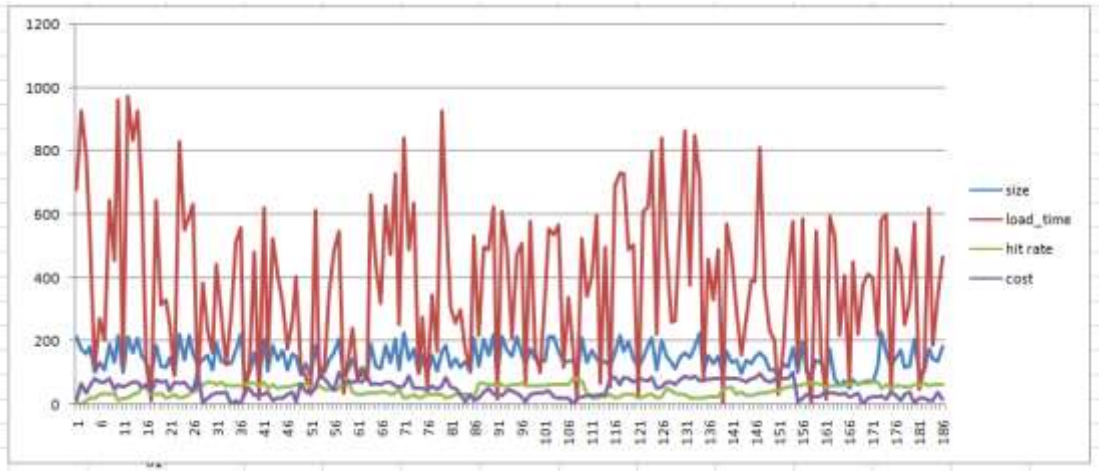
الشكل (13) مخطط حجم الصفحة والزمن ومعدل الإصابة لدى تطبيق تابع الاستبدال LRU

الشكل (14) يبين العلاقة بين حجم الصفحة وزمن تحميل الصفحة ومعدل الإصابة لدى تطبيق خوارزمية LFU



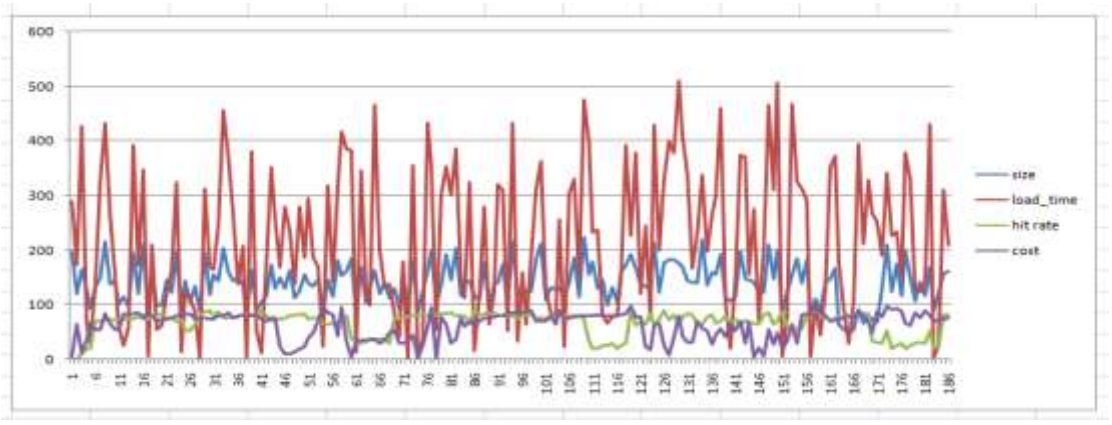
الشكل (14) مخطط حجم الصفحة وزمن التحميل ومعدل الإصابة لدى تطبيق خوارزمية LFU

الشكل (15) يبين العلاقة بين حجم الصفحة ومعدل الإصابة وزمن تحميل الصفحة لدى تطبيق خوارزمية Greedy dual مع تابع حساب كلفة المستند الذي تم شرحه أعلاه.



الشكل (15) مخطط حجم الصفحة وزمن التحميل ومعدل الإصابة لدى تطبيق خوارزمية Greedy Dual

الشكل (16) يبين العلاقة بين حجم الصفحة وزمن تحميل الصفحة ومعدل الإصابة لدى تطبيق خوارزمية GDFS.



الشكل (16) مخطط حجم الصفحة وزمن التحميل ومعدل الإصابة لدى تطبيق خوارزمية GDFS

من دراسة الأشكال السابقة نجد ما يلي:

- لاحظنا أنَّ أحجام الصفحات المدروسة بشكل عام تتراوح بين 45 كيلو بايت و 300 كيلو بايت،
- نموذج طلب المستخدمين للصفحات من ناحية الحجم أو الكلفة ذو طبيعة عشوائية عامة، إلا أنّه يأخذ شكلا دوريا من ناحية تكرار وحداثة الطلب بسبب طبيعة العمل في المرفأ التي يطلب المستخدم فيها معلومات مكثفة عن المانيفست والنظم المتصلة به لدى وصول السفينة، وتخف كمية الطلبات وتتحول لأغراض إحصائية بعد مغادرة السفينة والانتهاء من تحميل وإخراج البضاعة.
- كما يتبين لنا من الشكل (12) أنه قي حال غياب نظام كاش الويب، فإنّه توجد علاقة وثيقة بين حجم الصفحة وزمن تحميل المستند، وأكدّ ذلك قيمة معامل الارتباط correlation factor العالي بينها والمقاسة للأرقام الواردة في الشكل المذكور وهي 0.94141، ينخفض هذا المعدل لدى تطبيق خوارزميات الكاش بسبب تدخل الخوارزمية في تحديد البيانات التي سوف تحمّل من مصدرها الأصلي والبيانات التي سوف تحمّل من مخدم الكاش.
- أقصى زمن للتحميل كان بحدود 18 ثانية والزمن الأدنى تراوح بين ثلاث ثوان إلى خمس ثوان.
- زمن التحميل الوسطي للصفحات في حال غياب الكاش هو 9 ثوان.
- بالنسبة لجميع الخوارزميات بعد تطبيق الكاش تراوح زمن التحميل للصفحات بين 3 ثوان و 14 ثانية بحد أقصى وأكدت جميع الأرقام أنّ تطبيق خوارزمية كاش الويب يحسن أداء نظام المعلومات
- بالنسبة لخوارزمية LRU كان أقصى زمن للتحميل هو 10 ثوان ومتوسط زمن التحميل هو 5 ثوان، أما معدل الإصابة فكان بالمتوسط 53% أي نصف كمية المعلومات التي تم تسليمها للمستخدمين تم الحصول عليها من ذاكرة الكاش.
- بالنسبة لخوارزمية LFU كان أقصى زمن للتحميل هو 14 ثانية و متوسط زمن تحميل هو 8 ثوان ومعدل إصابة هو 40%
- بالنسبة لخوارزمية greedy dual فان أقصى زمن للتحميل هو 9 ثوان ومتوسط زمن التحميل هو 4 ثوان ومعدل إصابة 55% وبالنسبة لتابع الكلفة تراوح بين 2 و 99 بقيمة متوسطة 64.
- بالنسبة لخوارزمية GDFS أقصى زمن لتحميل الصفحة هو 4 ثوان و متوسط الزمن هو ثانيتين مع متوسط معدل إصابة 66% والكلفة تراوحت بين 2 و 97 بقيمة متوسطة 44.
- يعود انخفاض متوسط قيمة تابع الكلفة في خوارزمية GDFS إلى الأداء الأفضل للخوارزمية الذي أدى إلى الاحتفاظ بوحدات المعلومات ذات الكلفة العالية في الكاش، فانخفضت كلفة الدخل والخرج I/O cost وكلفة المعالجة CPU Cost الخاصة بمحرك قاعدة البيانات بسبب تحويل جزء من هذه العمليات إلى مخدم الكاش.

يمكن تلخيص ما سبق بالجدول رقم (2) المبين أدناه.

الجدول (2) ملخص القيم التي تم الحصول عليها من البيانات الوصفية التي قدمها نظام الكاش بعد تطبيقه باستخدام سياسات استبدال مختلفة.

avg cost	avg hit(%)	avg time(sec)	min time(sec)	max time(sec)	min size(KB)	avg size(KB)	max size(KB)	
--	--	9	5	18	46	197	275	Without Cache
--	53	5	3	10	55	144	290	LRU
--	40	8	2	14	51	140	227	LFU
64	55	4	2,5	9,7	49	143	277	Greedy Dual
44	66	2,2	2	4	55	140	280	GDFS

أسهم الكاش في تخفيض زمن تحميل الصفحة بنسب متفاوتة حسب تابع الاستبدال المطبق، ولكن جميع الخوارزميات المطبقة أعطت نتائج إجمالية أفضل من زمن تحميل الصفحة بغياب كامل لتقنية الكاش.

خوارزمية LRU أعطت نتيجة أفضل من خوارزمية LFU حيث كان معدل الإصابة أعلى وبقيم دورية عريضة مقارنة مع القيم الحادة لمعدل الإصابة لخوارزمية LFU، ويعود ذلك بعد مراجعة البيانات التي تمّ الاستعلام عليها ضمن المجال الزمني للبحث، إلى نموذج طلب مستخدمي نظام المانيفست، حيث يزيد الطلب بشكل دوري على معلومات معينة لدى وصول مانيفست السفينة إلى حين الانتهاء من إخراج البضاعة، ويقل الطلب على معلومات المانيفست تدريجياً إلى حين وصول مانيفست جديد، أمّا بالنسبة إلى الذرى الحادة لمعدل الإصابة في خوارزمية LFU فيعود ذلك إلى حصول طلبات مكثفة على أجزاء من المعلومات بسبب النظم الإحصائية ونظم استخراج التقارير والتي تقوم بعمليات مسح مكثفة ومتعددة للبيانات لاستخراج تقارير معينة حيث تحمّل هذه البيانات لفترة قصيرة نسبياً ومؤقتة في الكاش.

أعطت خوارزمية Greedy Dual أرقاماً جيدة فيما يخص زمن تحميل الصفحة ومعدل الإصابة، ولاحظنا أنّ ارتفاع معدل الإصابة وانخفاض زمن تحميل الصفحة يتبع تحميل وحدات ذات كلفة عالية إلى الكاش، ويعود ذلك إلى انخفاض الحمل على قاعدة البيانات ونظم الدخل والخرج بسبب تحميل الوحدات ذات الكلفة العالية إلى الكاش وتوفير المعالجة على مخدم قواعد البيانات.

أعطت خوارزمية GDFS الأداء الأفضل مقارنة مع باقي الخوارزميات، إذ انخفض زمن تحميل الصفحة بسبب وجود نسبة إصابة عالية في الكاش كما يبين المخطط، ويبين أنه تمّ الحصول على أجزاء كبيرة من وحدات المستندات من ذاكرة الكاش ويعود ذلك إلى إدخال عوامل الكلفة التي سمحت بالاحتفاظ بالوحدات المكلفة ضمن الكاش، بالإضافة إلى عامل تردد طلب المعلومة وحجم وحدة البيانات، وهو ما أدى إلى تخفيض الضغط على مخدمات قواعد البيانات بشكل أمثلي.

الاستنتاجات والتوصيات:

- اعتماد خوارزميات كاش الويب في النظم العاملة في مرفأ اللاذقية لتسريع أداء نظم استعادة المعلومات.
- خوارزمية *LRU* أعطت أداءً أفضل من خوارزمية *LFU* بسبب طبيعة الطلب الترددي للبيانات *Frequent Data Request*.
- خوارزمية *GDFS* أعطت نتائج أفضل من خوارزمية *Greedy Dual* بعد إدخال قياس وحدة البيانات وتردد طلبها من الكاش.
- إنّ طبيعة العمل المرفئي المعقدة وتعدد الأطراف التي تستخدم البيانات لأغراض مختلفة (تحميل البضائع – التفريغ – الوزن – الأبواب المرفئية – ...) تتطلب استخدام كافة المعلومات المتوفرة عن المستند من أجل تابع الاستبدال في خوارزمية الكاش، وبالتالي، فإنّ استخدام خوارزمية*GDSF* سمح بالحصول على الأداء الأفضل مقارنة مع الخوارزميات الأخرى من خلال التحديد الأفضل للملفات المطلوب إبقاءها في الكاش.
- القيام بتجزئة المعلومات في صفحات النظم المختلفة على شكل وحدات بيانات مرتبطة منطقيا بحيث تستطيع خوارزميات الكاش التعامل مع جزئيات المعلومات المخزنة واستخدامها في الصفحات المختلفة بدلاً من التعامل مع الصفحة بكاملها.

المراجع

[1]*Cisco VGlobal IP Traffic Forecast Update; 2010–2015, Presented by Mark Nowell – Senior Director Engineering, Cisco, IEEE 802.3 Bandwidth Advisory Ad-hoc, Sept* 2012.

[2]مديرية المعلوماتية في الشركة العامة لمرفأ اللاذقية

[*3*]El AbdouniKhayari, Rachid., Best, Michael.,Lehmann, Axel.,"Impact of Document Types on the Performance of Caching Algorithms in WWW", *Department of Computer Science University of The Armed Forces, Munich,2014.*

[*4*]Suresha.,Jayant, R.,"On Reducing Dynamic Web Page Construction Times", *Haritsa Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore,2014.*

[*5*]Iyengar, Arun., Challenger, Jim.,"Improving Web Server Performance by Caching Dynamic Data",*IBM Research Division, T. J. Watson Research Center.1997*

[*6*]Desai, Khyati .,"Bandwidth Optimization Using Content Aliasing OfProxy Server",1997.

[7]Shamsuddin, SitiMariyam; Ismail, Abdul Samad; Ali, Waleed ;" A Survey of Web Caching and Prefetching";Int. J. Advance. Soft Comput. Appl., Vol. 3, No. 1, March 2011 ,ISSN 2074-8523; Copyright © ICSRS Publication, 2011 www.i-csrs.org .

[8]Chauhan, Amit K. ; Chauhan, Vineet;" Exploring the Web Caching Method To Improve The Web Efficiency";International Journal of Computer Applications in Engineering Sciences VOL I, ISSUE IV , DECEMBER 2011 ISSN: 2231 -4946 Page 401.

[9]Plumley, Brandon Marcus;"An Investigation of Load Balancing in a

Distributed Web Caching System";A Thesis Submitted to the Committee on Graduate Studies in Partial Ful_llment of the Requirements for the Degree of Master of Science in the Faculty of Arts and Science TRENT UNIVERSITY Peterborough, Ontario, Canada Applied Modelling and Quantitative Methods M.ScProgram,January 2015.

[10]Moruz, Gabriel; Negoescu, Andrei; Neumann, Christian; W, Volker;" Engineering Efficient Paging Algorithms";Goethe University Frankfurt am Main. Robert-Mayer-Str. 11-15, 60325 Frankfurt am Main, Germany.2015

[11]Saemundsson, Trausti ;"An experimental comparison of cache algorithms"; Research Methodology, Reykjavik University, September 30, 2013

[12] Zheng, Dong ;"Differentiated Web Caching – A Differentiated Memory Allocation Model on Proxies";A thesis submitted to the School of Computing In conformity with the requirements for the Degree of Master of Science Queen's University Kingston, Ontario, Canada ,July 2004

[13]Tolosa, Gabriel; Feuerstein, Esteban;" Using Big Data Analysis to Improve Cache Performance in Search Engines";University of Buenos Aires, Argentina 2 National University of Lujan, Argentina tolosoft@unlu.edu.ar, efeuerst@dc.uba.ar AGRANDA 2015, 1º SimposioArgentino de GrandesDatos.

[14] Li, Jie; Arvidsson, Åke; Dán, György; Wu, Jinlong;" Performance Analysis of Local Caching Replacement Policies for Internet Video Streaming Services";Networking and Transmission Laboratory Acreo Swedish ICT AB, Kista, Sweden,2016.

[15] Timothy Y. Chow, Terence P. Kelly, Daniel M. Reeves,"Estimating Cache Hit Rates from the Miss Sequence", Enterprise Systems and Software Laboratory,HP Laboratories Palo Alto ,HPL-2007-155 September 17, 2007.