مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (37) العدد (3) 2015
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (37) No. (3) 2015

تخفيض زمن الوصول إلى ملفات الصور
باستخدام طريقة الفهرسة الشجرية المحسنة B^+ Tree

الدكتورة مريم ساعي [*]
الدكتورة كندة أبو قاسم [**]
أسامة محمد اسماعيل [***]

(تاريخ الإيداع 9 / 2 / 2015. قُبِل للنشر في 21/ 5 / 2015)

□ ملخّص □

يقترح البحث نظاماً جديداً يهدف إلى تخفيض زمن البحث عن ملفات الصور images عن طريق اقتراح آلية فهرسة جديدة تعالج العيوب التي عانت منها خوارزميات الفهرسة المستخدمة بحيث يصبح زمن الوصول لهذه الملفات أقل ما يمكن.

تم بدايةً في هذه الورقة توضيح أهمية الأرشفة Archiving في تنظيم الملفات عن طريق تصميم قاعدة بيانات Database وتخزين ملفات الصور فيها وتسجيل الأزمنة اللازمة للحصول على الملفات المطلوبة من قاعدة البيانات، بعد ذلك تم إجراء عملية الفهرسة Indexing لملفات الصور المخزنة في قاعدة البيانات عن طريق اقتراح خوارزمية جديدة – B^+ Tree المحسنة – تهدف إلى تنظيم ملفات الصور وفق آلية معينة تسهل الوصول للملفات المطلوبة وتم إجراء عمليات الاستعلام queries وتسجيل الأزمنة المستغرقة من أجل مقارنتها مع الأزمنة اللازمة للوصول إلى الملفات قبل الفهرسة بهدف إظهار كفاءة الطريقة المقترحة.

الكلمات المفتاحية: الفهرسة الشجرية، قواعد البيانات، الفهرسة، الاستعلام

[*] أستاذ مساعد – قسم هندسة الحاسبات والتحكم الآلي – كلية الهندسة الميكانيكية والكهربائية – جامعة تشرين – اللاذقية – سورية
[**] أستاذ مساعد – قسم هندسة الحاسبات والتحكم الآلي – كلية الهندسة الميكانيكية والكهربائية – جامعة تشرين – اللاذقية – سورية
[***] قائم بالأعمال – قسم هندسة الحاسبات والتحكم الآلي – كلية الهندسة الميكانيكية والكهربائية – جامعة تشرين – اللاذقية – سورية

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (37) العدد (3) 2015
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (37) No. (3) 2015

Access Time Reduction To Image Files Using Enhanced B$^+$ Tree Indexing Method

Dr. kinda aboukassem[*]
Dr. Mariam saii[**]
Usama Esmaeel[***]

(Received 9 / 2 / 2015. Accepted 21 / 5 / 2015)

□ ABSTRACT □

The Research suggests a novel model aims to reduce the time of search for image files by proposing a new indexing mechanism to avoid the plague algorithm used with indexing so that the access time to these files becomes as less as possible.

The first stage in this paper is to clarify the importance of archiving in organizing files via designing a database, storing images in it and recording the times needed to obtain the required files from the database. Then the indexing process for image files stored in the database is applied by proposing a new algorithm -B+ Tree enhanced- for organizing image files according to a certain mechanism to facilitate accessing any file, conducting queries and recording the times used to get those files from the database to compare them with the times required to access files before indexing in order to show the efficiency of the proposed method.

Key words: Tree indexing, Database, Indexing, query, B$^+$ Tree

[*]**Associate Professor in computer and automatic control engineering, faculty of Mechanical and electrical engineering, Tishreen university, Lattakia, Syria.**
[**]**Associate Professor in computer and automatic control engineering, faculty of Mechanical and electrical engineering, Tishreen university, Lattakia, Syria.**
[***]**Academic Assistant, Department of computer and automatic control engineering, faculty of Mechanical and electrical engineering, Tishreen university, Lattakia, Syria.**

مقدمة :

تعتبر الفهرسة ذات أهمية عالية في قواعد البيانات لكونها تؤمن الوصول السريع لنتيجة الاستعلام ضمن قاعدة البيانات، وتتجلى أهمية ذلك في عمليات أرشفة الملفات files من خلال قواعد البيانات، فعمليات البحث عن الملفات تحتاج لأزمنة كبيرة تتفاوت تبعاً لعدد الملفات وحجم القرص الصلب.

قام عدد من الباحثين في جامعة سنغافورة الوطنية بنشر بحث في عام 2010 حيث اقترح عدة خوارزميات بالفهرسة ومنها الفهرسة الشجرية R–Tree، وكانت أزمنة الاستعلام عن البيانات تزداد بشكل كبير مع زيادة حجم البيانات. **[1]**

قدم البروفيسور Chris Tseng عام 2007 في قسم علوم الحاسب في جامعة San Jose بأمريكا بحثاً علمياً تطرق فيه إلى تطبيقات الإنترنت وطرق التعامل مع كمية كبيرة من البيانات المرتبطة بنوع التطبيق، كما قدم هذا البحث طرقا للتعامل مع البيانات الغامضة كالمنطق الضبابي والشبكات العصبونية. **[2]**

أيضاً في عام 2007 قام الباحثان Rui Li, Bir Bhanu بنشر مقالة في مجلة Elsevier جامعة California مركز البحوث للأنظمة الذكية في أمريكا، وقد تعامل الباحثان فيه مع قواعد البيانات الموزعة، وتخصص بدراسة حالات عدم الوثوقية uncertain ضمن قواعد البيانات المكانية أو الجزئية كما تم نمذجة حالات عدم الوثوقية باستخدام دالة الكثافة الاحتمالية وبناء آلية فهرسة لها بالاعتماد على مخطط غاوص (OGMH) لدعم البيانات المحددة وغير المحددة، كما تضمن البحث أيضاً نمذجة للاستعلامات المحددة وغير المحددة،وحالات عدم الموثوقية فيها. **[3]**

نتيجة الإطلاع على الأبحاث السابقة في هذا المجال يمكن تلخيص العيوب التي عانت منها على الشكل الآتي:

1 –أزمنة البحث تتناسب طرداً مع عدد الملفات.

2 –التطبيقات المستخدمة بحاجة إلى قوانين خاصة أثناء عملية البناء ولا يمكن أن يكون المستخدم في معظم الأحيان على دراية جيدة بهذه القوانين.

3 –بعض خوارزمية الفهرسة المتبعة لا تعيد تنظيم نفسها بشكل أوتوماتيكي عند حدوث تغيرات بسيطة كالحذف Deletion أو الإضافة insertion، بالتالي هي بحاجة لتنظيم دوري للملفات مما يقلل الأداء.

4 –في الفهرسة الشجرية فان جميع المسارات من الجذر إلى الأوراق ليست واحدة وبالتالي حصول أزمنة مختلفة عند البحث عن ملفات ذات مسارات مختلفة.

5 –أزمنة عمليات الإدخال(تخزين الملفات) تتزايد مع تزايد عدد الملفات.

6 –أزمنة الاستعلام تزداد بزيادة عدد الملفات.

بناءاً على ما سبق اقترح البحث آلية فهرسة جديدة تسمى الفهرسة الشجرية B^+ tree المحسنة بحيث يتم الاعتماد على الفهرسة الشجرية B^+ tree وإجراء تعديل على بنية الشجرة بهدف تحسين عملية الفهرسة والإبقاء على التميز الخاص بها.

ولتحليل هذا الموضوع بآلية عملية سنصمم قاعدة بيانات بدون عملية فهرسة، وسنجري عليها الاختبارات المناسبة التي توضح أهمية عملية الفهرسة، وهذا يتطلب اختيار تطبيق يوضح بشكل جيد دور الأرشفة في عملية الوصول إلى الملفات بآلية سريعة، بالإضافة إلى دور الفهرسة في عملية إظهار نتيجة الاستعلام بأقصر زمن.

تم اختيار قاعدة بيانات لذاتيات مواطنين مع صور خاصة بهم كتطبيق لشعبة تجنيد، وهذا التطبيق جيد لكونه يحتوي على ملفات صور خاصة بكل مواطن، وهذا يحتاج لعملية أرشفة للملفات بالإضافة لكون عملية الأرشفة هذه تحتاج لعملية فهرسة ضمن قواعد بيانات، وبالتالي سنعمل على تصميم قاعدة البيانات من خلال برنامج SQL Server لكونه بيئة جيدة لبناء قواعد البيانات، وتدعم حزمة عريضة من الاستعلامات التي يمكن أن نحتاجها، وسنعمل على تصميم واجهة مستخدم من خلال بيئة برمجة تسهل الاتصال والتعامل مع قاعدة البيانات بالإضافة إلى سهولة إجراء الاختبارات واستحصال النتائج.

أهمية البحث وأهدافه

يهدف هذا البحث إلى اقتراح نموذجاً جديداً لفهرسة ملفات الصور مع الأخذ بعين الاعتبار تقليل زمن البحث بحيث يصبح أقل ما يمكن باستخدام الفهرسة الشجرية المحسنة B^+ Tree وذلك بمساعدة بيئة الـ net.، ومن ثم بناء استعلامات queries جيدة باستخدام تعليمات الـ SQL بهدف تحويل عملية البحث عن الملفات إلى عملية استعلام query ضمن قاعدة البيانات Database كونها تؤمن الوصول السريع للبيانات.

مشكلة البحث

نتيجة اتساع المعلومات وزيادة حجمها ظهرت الحاجة إلى إدارة كم هائل من المعلومات، وقد تبين عجز طرق البرمجة التقليدية عن مواكبة التطور الحاصل في حجوم المعطيات أدى ذلك إلى البحث عن طرق أخرى تهدف في المقام الأول إلى تحسين وتسهيل طرق الوصول السريع إلى البيانات data باستخدام خوارزمية من خوارزميات الفهرسة.

طرائق البحث و مواده

تم تقسيم العمل إلى عدة مراحل حيث اقترح البحث بدايةً بناء تطبيق للبحث عن ملفات الصور ذات التواجد العشوائي ضمن القرص الصلب للحاسوب واختبار هذا التطبيق بهدف توضح أهمية الأرشفة في تنظيم الملفات بالإضافة إلى الأزمنة المستهلكة في عمليات البحث ودور تزايد عدد الملفات في زيادة زمن الوصول للبيانات.

في المرحلة الثانية اقترح البحث بناء قاعدة بيانات وإدخال البيانات إليها بدون فهرسة، ومن ثم إجراء تحليل لقاعدة البيانات هذه من حيث الأزمنة المستغرقة للحصول على الملفات المطلوبة عبر مجموعة من الاستعلامات بهدف توضيح أهمية الفهرسة في جدولة الملفات. وفي المرحلة الثالثة قدم البحث مجموعة من الخوارزميات لاختيار الخوارزمية المناسبة من حيث مبدأ العمل واقتراح آلية فهرسة جديدة تتلافى المشاكل التي تعاني منها خوارزميات الفهرسة المستخدمة.

أخيرا اقترح البحث تصميم قاعدة بيانات وفقاً للبنية المقترحة لآلية الفهرسة المستخدمة و إجراء تحليل لقاعدة البيانات هذه من خلال إجراء نفس الاختبارات المطبقة على قاعدة البيانات وتحديد الأزمنة المستغرقة للوصول إلى البيانات، ومن ثم إجراء مقارنة للنتائج المحققة من خلال البنية المقترحة.

1- الأرشفة Archiving

تتطلب عمليات إيجاد ملف محدد ضمن القرص الصلب أو وحدات التخزين بشكل عام إلى آلية بحث، وآليات البحث هذه محددة من قبل نظام التشغيل بالإضافة إلى التعليمات الأساسية الخاصة بالمعالج المستخدم، ولكن بشكل

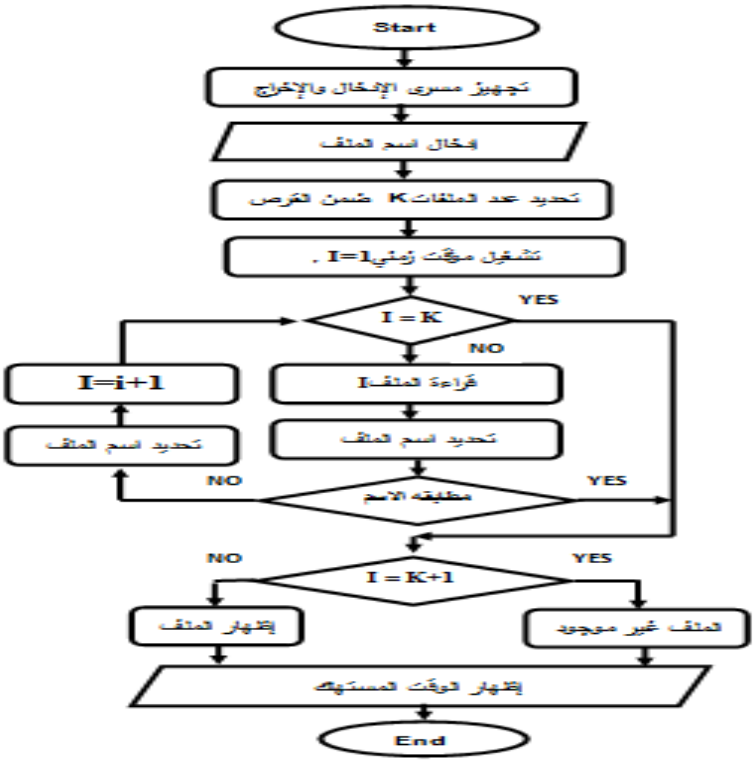
عام عمليات البحث هذه تعتمد على ترتيب هذه الملفات من خلال بنية هرمية وإجراء علميات مقارنة لأسماء هذه الملفات مع الملف المطلوب.

1-1- دور الأرشفة في عملية البحث عن الملفات

قدم العديد من الباحثين مجموعة من الخوارزميات اعتمد قسم منها على خوارزميات فرز تسبق عمليات البحث وغيرها من الآليات، لكنها بمجملها غير قادرة على تحقيق نتائج مقنعة لكون الملف المطلوب ذو تواجد عشوائي ضمن القرص، وقد يكون غير موجود أصلاً وهي الحالة الأسوأ والتي تكلف أكبر زمن للوصول للبيانات.

1-2- تصميم خوارزمية البحث عن الملفات

اقترح البحث بناء تطبيق يقوم بعملية البحث عن ملف ضمن سواقة محددة من القرص الصلب، ويحقق آلية حساب للزمن المستغرق للوصول إلى البيانات وعرضها. عملية تصميم مثل هذه الواجهة يمكن تحقيقه بسهولة من خلال بيئة البرمجة net. والتي تحقق آلية سهلة من حيث التصميم والاستخدام. نحتاج لبناء مسار إدخال وإخراج من أجل عملية قراءة الملفات، ومن ثم نحتاج لعملية تحليل لأسماء الملفات من أجل المطابقة في عملية البحث مع اسم الملف المطلوب، ومن ثم عرض الملف، وهذه الملفات التي تم البحث عنها هي ملفات صور، وفي نهاية العمل قمنا بعرض الزمن المستهلك لتحقيق هذه العملية، حيث تمت برمجة هذه الخوارزمية من خلال بيئة net.، وتم تجريب هذه العملية على عدة ملفات لا على التعيين لتحديد الأزمنة المستهلكة في كل من حالتي تواجد الملف أو عدم تواجده. يبين الشكل (1-4) عمل خوارزمية برنامج البحث عن الملفات:



الشكل (1-4) خوارزمية برنامج البحث عن الملفات

2- الفهرسة وأنواعها

تعتبر الفهرسة ذات أهمية عالية في قواعد البيانات لكونها تؤمن الوصول السريع للبيانات المراد فهرستها ضمن قاعدة البيانات، وتتجلى أهميتها في عمليات أرشفة الملفات من خلال قواعد البيانات، فعمليات البحث عن

الملفات تحتاج لأزمنة كبيرة تتفاوت تبعاً لعدد الملفات وحجم القرص الصلب، حيث يتم ترتيب البيانات وفقا لآلية معينة بحيث يتم الوصول إليها أقل زمن ممكن من خلال بناء برامج تطبيقية. **[4]**

2-1- دراسة تحليلية لآليات الفهرسة وتوضيح الآلية المقترحة

تنوعت آليات الفهرسة المطبقة والمستخدمة وتباينت هذه الآليات من حيث الميزات والمساوئ، نذكر منها على سبيل المثال B^+ Tree Index، Ordered Indices، Sparse Index، Multilevel Index، Static Hash Indices، Dynamic Hash Indices، B-Tree Index . **[5]**

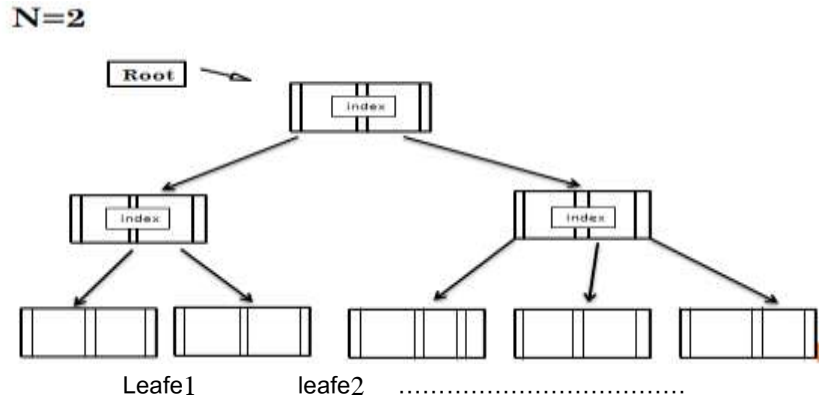
عملية اختبار آلية الفهرسة تتم من خلال عدة نقاط أهمها زمن الوصول arrival time وكذلك زمن الإدخال insertion time و زمن الحذف Deleting time والحجز المسبق Space Overhead والتي من خلالها يتم اختبار الخوارزمية المراد استخدامها في عملية الفهرسة بهدف إعطاء أفضل النتائج. **[6]**

2-2- دراسة خوارزمية الفهرسة B^+ Tree:

تستخدم هذه الطريقة بشكل واسع لأنها تعيد تنظيم نفسها بشكل آلي عند حدوث تغيرات بسيطة كالحذف deletions والإضافة insertions، كما أن إعادة تنظيم الملفات المدخلة لا يتطلب المحافظة على الأداء أما الكلفة فتزداد بزيادة عمليات الإدخال والحذف، وينقص الأداء بنمو الملف، وبالتالي تتطلب هذه الطريقة تنظيماً دورياً للملفات المدخلة. **[7]**

يمكن التعبير عن هذه الطريقة B^+ Tree بشجرة لها جذور، وتتميز بأن كل المسارات من الجذر إلى الأوراق هي بنفس الطول، وكل عقدة لا تعتبر جذراً للشجرة، فإنها تملك بين n و n/2 ابن، كما أن عقدة الورقة تملك قيم بين (n/2-1) و (n-1) مفتاح بحث، أما إذا كان الجذر ورقة (لا يوجد عقد أخرى في الشجرة)، فانه يملك قيماً بين 0 و (n-1)قيمة. **[8]** .

يبين الشكل (2-4) البنية العامة لشجرة الفهرسة B^+ Tree من أجل (n=2) : **[9]**



الشكل (2-4) البنية العامة للشجرة B^+ Tree

بناءً على آلية عمل هذه الخوارزمية يمكن تلخيص عدة نتائج حول هذه الخوارزمية هي:

1 -تتطلب عمليات الإدخال أزمنة متفاوتة تتزايد مع تزايد الملفات المؤرشفة.

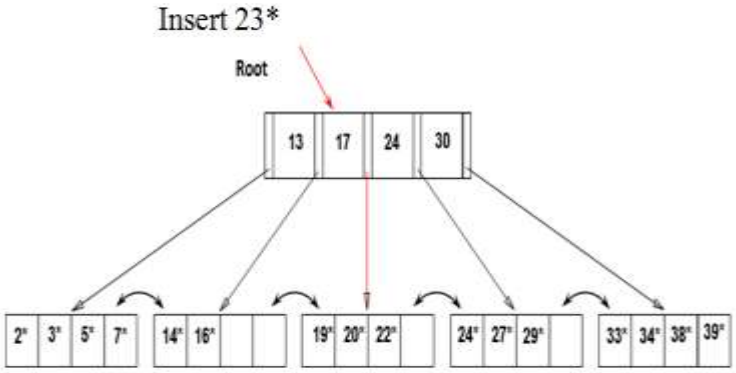
2 -عمليات الحذف ضمن البنية الشجرية للخوارزمية تتطلب أزمنة إعادة ترتيب تتزايد مع تزايد عدد الملفات

3 -تتميز هذه الخوارزمية بأزمنة متوازنة لعمليات الاستعلام أي عملية البحث عن أي ملف تتطلب نفس الزمن مهما كبر عدد الملفات المؤرشفة.

4 -مساحات الحجز تعتبر جيدة لكون عملية التفريع تتطلب امتلاء نصف عدد عناصر العقدة أو أكثر لأن عدد الفراغات الشاغرة الأعظمي يرتبط بعدد العقد node والأوراق leaf.

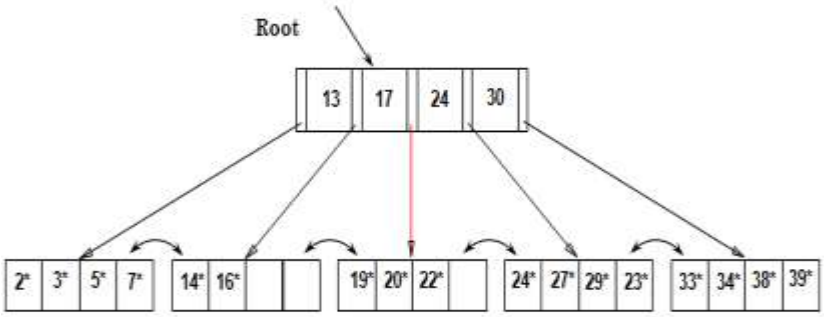
2-3- مساوئ الفهرسة الشجرية B^+ Tree

بفرض لدينا قاعدة بيانات مفهرسة بالطريقة الشجرية (n=4) ونريد إدخال بيانات، تكزن الشجرةكما هو موضح بالشكل (3-4):



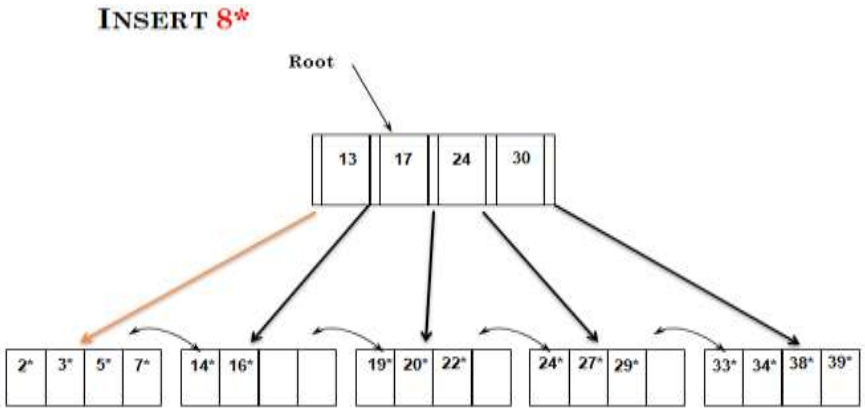
الشكل (3-4) شجرة B^+ Tree قبل إدخال البيانات

وبعد إدخال البيانات تصبح الشجرة كما هو موضح بالشكل (4-4):



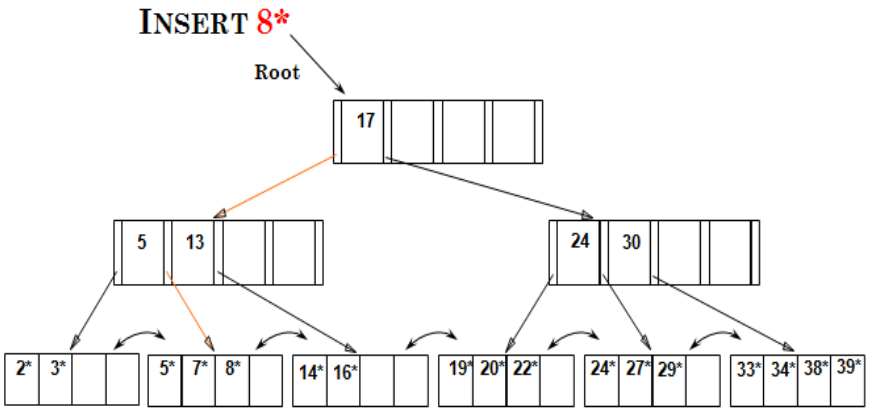
الشكل (4-4) شجرة B^+ Tree بعد إدخال البيانات

نلحظ من الشكل أنه عندما يكون هناك مكان للبيانات المدخلة بأحد أوراق الشجرة فانه لا حاجة للتجزئة، وفي حال لا يوجد لها مكان تكون الشجرة كما في الشكل (5-4) :



الشكل (4-5) شجرة B+ Tree قبل إدخال البيانات

وبعد إدخال البيانات تصبح الشجرة كما هو موضح بالشكل (4-6): **[10]**



الشكل (4-6) شجرة B+ Tree بعد إدخال البيانات

نلحظ عند إدخال البيانات إلى قاعدة بيانات مفهرسة بالطريقة الشجرية، ولا يوجد لها مكان فإنه يتم تجزئة الورقة إلى ورقتين لتصبح الورقة السابقة عقدة مع وضع مؤشر ليدل على مكان توضع البيانات. لهذا اقترح البحث تحسين طريقة الفهرسة الشجرية B+ Tree عن طريق إجراء بعض التغيرات على الشجرة بحيث تصبح كل المسارات من الجذر إلى الأوراق واحدة، مع الحفاظ على التميز الخاص بالشجرة.

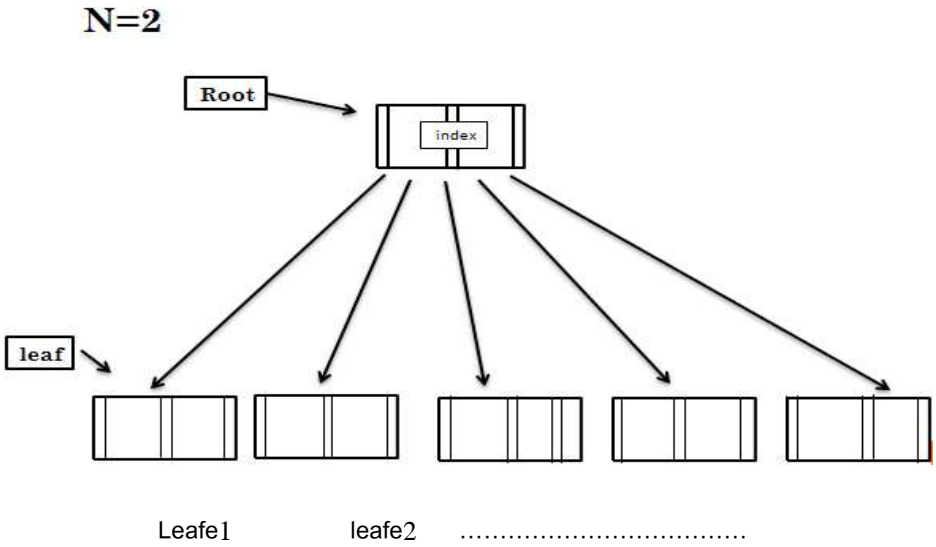
3- خوارزمية الفهرسة المقترحة:

على الرغم من تميز خوارزمية B+ Tree في أزمنة الوصول إلى أنها تعاني من استهلاك أزمنة كبيرة في عمليات الإدخال والحذف، لذلك نحن بحاجة إلى آلية فهرسة تختصر بشكل جيد أزمنة الإدخال والحذف أو تجعل هذه الأزمنة مستقلة نسبياً عن عدد الملفات المؤرشفة.

يمكن تحقيق الاختصار في أزمنة الإدخال والحذف من خلال عملية تغير في البنية الهرمية لخوارزمية B+ Tree، وذلك بجعل هذه البنية ذات مستوى واحد فقط هو مستوى الأوراق Leafs فقط، بحيث تخصص كل ورقة لمجموعة محددة من الملفات وبالتالي فعملية الإدخال هنا تتم مباشرة للمكان المخصص للملف، والذي لا يمكن أن

يتغير مع تزايد عدد العناصر، وبالتالي زمن الإدخال سيصبح ثابتاً، وكذلك الأمر بالنسبة لزمن الحذف، أما بالنسبة لزمن الوصول فيكون أقل ما يمكن هو الزمن الخاص بعملية البحث ضمن الورقة الواحدة.

يبين الشكل (7-4) البنية العامة للشجرة B^+ Tree المعدلة أو المحسنة:



الشكل (7-4) البنية العامة للشجرة B^+ Tree المعدلة

تعتمد خوارزمية الفهرسة بشكل أساسي على آلية بناء وتنظيم قواعد البيانات، وهنا تمثل كل ورقة لخوارزمية الأرشفة جدول ديناميكي يتم بناؤه عند الحاجة، وهذه العملية تتم من خلال اعتماد آلية بناء جداول ديناميكية خاصة بالعناصر المؤرشفة مع وجود جدول واحد ثابت هو الجدول الخاص بفهرسة الجداول الديناميكية.

النتائج والمناقشة

1-اختبار برنامج البحث

تم اختبار برنامج البحث المصمم من خلال تنفيذ عمليات بحث عن ملفات عشوائية، وقد تم توزيع هذه الاختبارات على الأقسام التالية:

1 -الأزمنة المستهلكة للبحث عن عدة ملفات ضمن عدد محدد من الملفات.

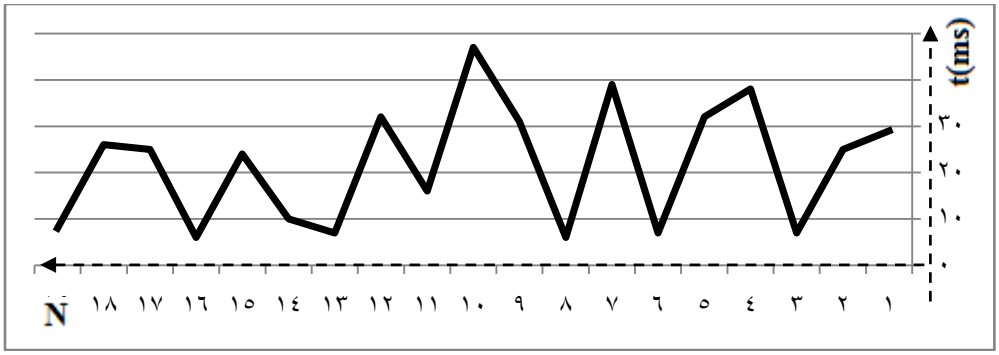
2 -الأزمنة المستهلة في البحث عن ملف من أجل أعداد متزايدة للملفات.

1-1- الحالة الأولى:الأزمنة المستهلكة للبحث عن عدة ملفات ضمن عدد محدد من الملفات:

لتنفيذ هذا الاختبار تم تخزين عدد محدد من ملفات الصور ضمن مسار واحد، وتم إجراء عمليات البحث عن ملفات بأسماء معينة، وكانت الأزمنة المستهلكة في عمليات البحث هذه موزعة كما في الجدول(1) حيث t(ms) هو زمن البحث بالميلي ثانية، n عدد ملفات الصور. يبين الجدول(1) أزمنة البحث بعدد ثابت من الملفات:

N=45																			
8	26	25	6	24	10	7	32	16	47	31	6	39	7	32	38	7	25	29	t(ms)

يمكن تحليل هذه النتائج من خلال رسم المخطط البياني للجدول السابق، كما هو مبين بالشكل (1-5):



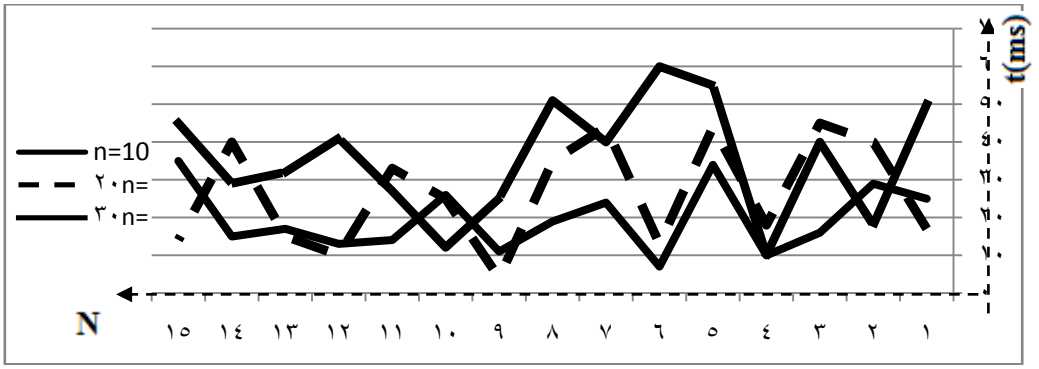
الشكل (1-5) مخطط تابعية زمن البحث من أجل عدد محدد من الملفات

1-2- الحالة الثانية: الأزمنة المستهلكة للبحث عن ملف ضمن عدد متزايد من الملفات

لتنفيذ هذا الاختبار قمنا بتخزين عدد محدد من ملفات الصور، وبدأنا عمليات البحث عن ملفات معينة، ومن ثم قمنا بزيادة عدد الملفات لنعيد عملية البحث من جديد فكانت الأزمنة المستهلكة موزعة كما هو مبين بالجدول الجدول(2):

عدد الملفات N=10															
35	15	17	13	14	26	11	19	24	7	34	10	16	29	25	t(ms)
عدد الملفات N=20															
14	40	15	10	33	25	4	35	44	13	44	18	45	40	17	t(ms)
عدد الملفات N=30															
45	29	32	41	27	12	25	51	40	60	55	10	40	18	50	t(ms)

يمكن تحليل هذه النتائج من خلال رسم المنحني البياني الموضح في الشكل (2-5):



الشكل (2-5) تابعية زمن البحث من أجل عدد متزايد من الملفات

من خلال المنحنيات الناتجة عن الاختبارات السابقة نجد النتائج التالية:

1 - أزمنة البحث أزمنة متفاوتة تبعاً لترتيب الملف ضمن قرص التخزين.

2 – عمليات البحث عن ملف غير موجود يتطلب الزمن الأعظمي لعملية البحث.

3 – الزمن الأعظمي لعملية البحث يتزايد مع تزايد عدد الملفات.

من خلال هذه النتائج نلحظ تزايد أزمنة البحث عن ملفات الصور بزيادة عددها من جهة وبمسار تخزينها من جهة أخرى كما أن أزمنة البحث الناتجة هي قيم عشوائية وهذا عائد لمسارات التخزين العشوائية المتواجدة فيها تلك الملفات، لهذه الأسباب تتجلى أهمية أرشفة الملفات بواسطة قواعد البيانات بحيث يتم تخزين أسماء الملفات بالإضافة إلى المسارات الخاصة بهذه الملفات بحيث تتحول عملية البحث عن الملف إلى عملية استعلام ضمن قاعدة البيانات تؤمن سهولة الوصول إلى الملفات المطلوبة.

2- آلية بناء قاعدة البيانات

لا يحتاج هذا التطبيق إلى قاعدة بيانات معقدة، وإنما فقط جدول واحد يحتوي على المعلومات الشخصية الأساسية بالإضافة إلى مسار الصورة الشخصية ضمن القرص الصلب، وبالتالي أي عملية بحث عن صورة شخصية تتم من خلال استعلام ضمن قاعدة البيانات تبعاً لاسم الشخص أو الرقم الوطني بحيث تكون نتيجة الاستعلام هي المسار المناسب للصورة ضمن القرص الصلب، أي تحولت عملية البحث ضمن القرص الصلب عن الملف المناسب إلى استعلام ضمن قاعدة البيانات. يمكن بناء قاعدة البيانات هذه من خلال برنامج SQL Server وقد اخترنا الحقل image_id كمفتاح أساسي للجدول لكونه رقم فريد يعبر عن الرقم الوطني.

3- هيكلية بناء تطبيق الأرشفة:

يمكن توزيع العمل ضمن تطبيق الأرشفة على عدة محاور كما يلي:

- بناء الاتصال مع قاعدة البيانات.
- استعراض الملفات على الحاسب لتحديد الصورة الشخصية.
- إدخال المعلومات الشخصية مع توابع التأكد من صحة هذه المعلومات.
- إضافة السجلات إلى قاعدة البيانات.
- الاستعلام عن سجل تبعاً للرقم الوطني.
- الاستعلام عن سجل حسب الاسم الشخصي.
- حذف سجل.
- تابع حساب الزمن المستهلك من قبل الاستعلام. **[11]**

3-1- اختبار برنامج الأرشفة

لتوضيح أهمية الفهرسة سنجري اختبارات على الأزمنة المستهلكة في الاستعلام لبرنامج الأرشفة بدون عملية الفهرسة، ويمكن تحقيق العديد من الاختبارات تصنف إلى:

1 -اختبارات الأزمنة من أجل ملفات بمسارات متعددة من أجل عدد محدد من السجلات.

2 -اختبار الأزمنة ضمن مسار محدد من أجل عدد متزايد من السجلات ضمن قاعدة البيانات.

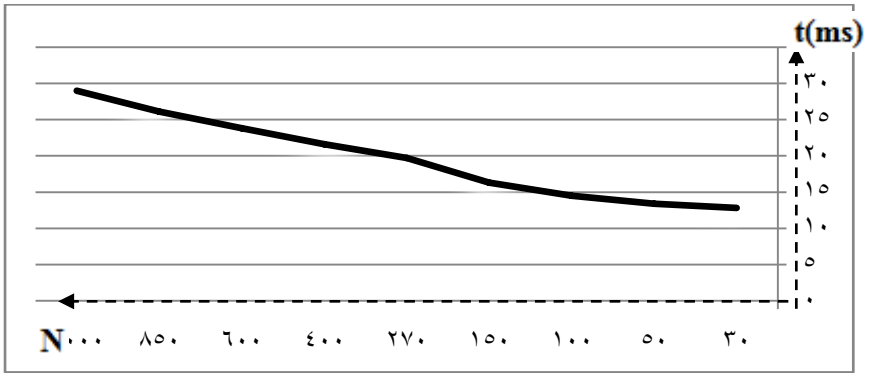
3 -الأزمنة المستهلكة ضمن مسارات متعددة من أجل عدد متزايد من السجلات.

يعتبر الاختبار الثاني هو الاختبار الأهم لكونه يعبر عن الحالة الطبيعية لقواعد البيانات من حيث تزايد عدد السجلات بالإضافة لكونه يحقق أفضل حالات لعملية عرض الملف لكون الملفات مخزنة ضمن مسار واحد، لذلك

أجرينا هذه الاختبار مع تزايد تدريجي للسجلات ضمن قاعدة البيانات فحصلنا على النتائج الموضحة في الجدول(3) حيث يبين أزمنة البحث بعدد متزايد من السجلات:

n	30	50	100	150	270	400	600	850	1000
t(ms)	12.8	13.4	14.5	16.3	19.7	21.6	23.8	26.1	29

يمكن توضيح هذه النتائج من خلال مخطط بياني يسهل عملية استخلاص النتائج والشكل (3-5) يوضح منحني تغير زمن البحث بزيادة عدد السجلات:



الشكل (3-5) تابعية زمن البحث بزيادة عدد السجلات

من خلال المنحني البياني نلاحظ تزايد أزمنة البحث عن الملفات بزيادة عددها أي أن العلاقة بين الزمن المستهلك وعدد السجلات – الملفات – عبارة عن تابع أسي يمكن التعبير عنه بصيغة رياضية تقريبية مبينة بالعلاقة :(1)

$$T = e^{ax} + b \qquad (1)$$

حيث a : ثابت يتعلق بعدد السجلات.

b : ثابت يتعلق بالزمن الخاص بعمليات عرض الملف.

من خلال هذا التحليل يمكن أن نخلص إلى النتائج التالية:

1 عملية الأرشفة للملفات لا تعتبر كافية لعملية توفير زمن البحث.

2 أزمنة البحث عن الملفات تتزايد مع تزايد عدد السجلات.

3 ترتبط أزمنة الاستعلام عن الملفات من خلال قواعد بيانات الأرشفة بدون الفهرسة بعلاقة أسية مع العدد الأعظمي للملفات المخزنة ضمنها.

4 عملية الأرشفة تحقق إزاحة في للتابع الذي يربط عدد السجلات بالزمن دون تغيير في شكل التابع.

إن تزايد أزمنة البحث يمكن ملاحظتها بشكل أكبر عند تزايد عدد الملفات بشكل كبير جداً عندها فان الوصول إلى الملف المطلوب يستغرق وقتاً كبيراً، لذلك اقترح البحث آلية فهرسة تقلل زمن البحث ويحقق الاستعلامات بتابع مناسب بين عدد السجلات والزمن اللازم للوصول إلى الملفات. ونظراً لكون عملية البناء للجداول عملية ديناميكية، لذلك نحتاج لبناء صنف ضمن لغة البرمجة خاص بعملية البناء الديناميكي، يتضمن كل العمليات اللازمة للتعامل مع الجداول من عمليات إدخال أو حذف أو استعلام. تم الاعتماد على هذا الصنف في تصميم تطبيق ضمن بيئة net. لإجراء عمليات الاختبار على الخوارزمية المصممة.

4- اختبار خوارزمية الفهرسة المصممة

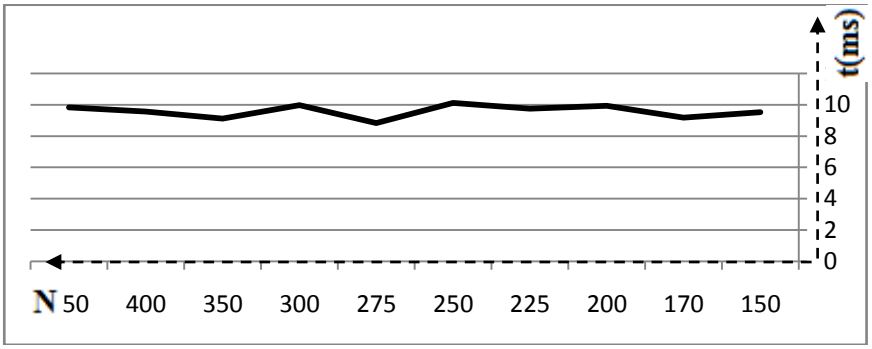
تم إجراء اختبارات مماثلة للاختبارات السابقة وكانت النتائج في الحالتين التاليتين على الشكل:

4-1- الحالة الأولى: اختبار أزمنة البحث ضمن مسار واحد مع تزايد عدد الملفات

يفيد هذا الاختبار في توضيح أن تزايد عدد الملفات لا يؤثر على أزمنة البحث، وهذا ما توضح من خلال النتائج التي حصلنا عليها سابقا. تم تكرار الاستعلام لعدة ملفات من أجل كل قيمة لعدد الملفات الكلي فحصلنا على النتائج كما في الجدول(4) حيث يبين أزمنة البحث بعدد متزايد من الملفات:

450	400	350	300	275	250	225	200	170	150	n
9.82	9.56	9.11	9.96	8.83	10.1	9.73	9.91	9.17	9.52	t(ms)

من خلال النتائج نلاحظ أن القيمة الوسطية لأزمنة الاستعلام هي 9.5ms وهذا ما يوضحه الشكل(4-5):



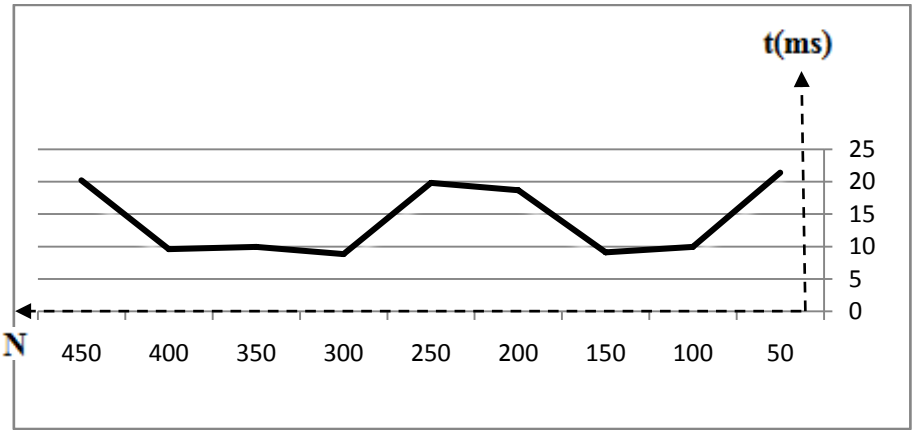
الشكل (4-5) تابعية زمن البحث مع تزايد عدد الملفات

4-2- الحالة الثانية: اختبار أزمنة البحث ضمن مسارات متعددة مع تزايد عدد الملفات

من خلال هذا المسار سنوضح دور تنوع مسارات الملفات في زمن البحث من خلال النتائج التي حصلنا عليها والمبينة في الجدول (5) حيث يبين أزمنة البحث بعدد متزايد من الملفات:

450	400	350	300	250	200	150	100	50	n
20.24	9.61	9.96	8.83	19.82	18.7	9.12	9.92	21.4	t(ms)

نلحظ أن تعدد المسارات سبب زيادة زمن الاستعلامات وبالتالي أصبحت القيمة الوسطية 15ms وهذا ما يوضحه الشكل (5-5):



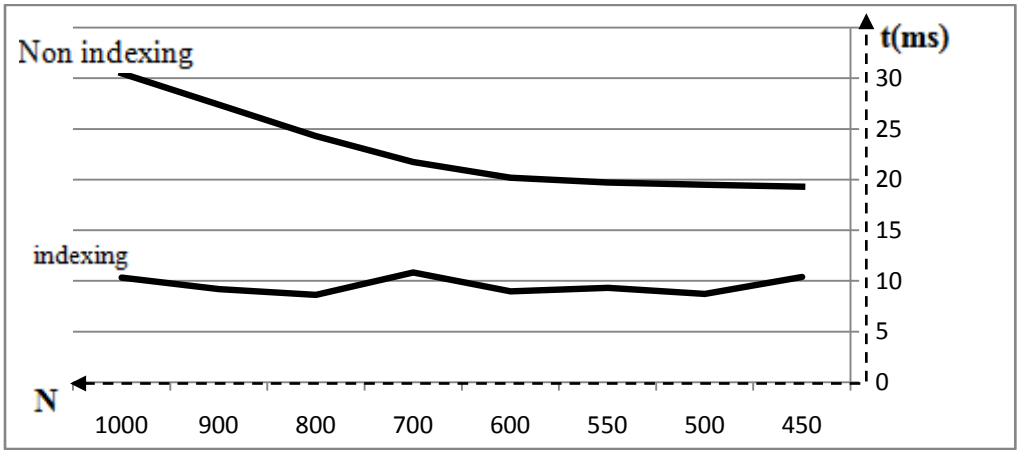
الشكل (5-5) تابعية زمن البحث مع تزايد عدد الملفات من أجل عدة مسارات للملفات

نظراً لكون نتيجة البحث عن الملفات من خلال الخوارزمية المصممة حققت نتائج جيدة من حيث ثبات الزمن لاستعلامات البحث، وبالتالي لسنا بحاجة للمقارنة إلا مع عملية البحث بدون عملية فهرسة لتأكيد أهميتها من حيث ثبات الزمن والتوفير الحاصل في الأزمنة.

بالعودة للنتائج التي حصلنا عليها سابقا ومقارنتها مع النتائج التي حصلنا عليها من ضمن هذه الفقرة يمكن مقارنة النتائج وتحديد الفوائد المحققة من خلال الخوارزمية المصممة ضمن هذا البحث. يوضح الجدول(6) أزمنة البحث قبل وبعد الفهرسة:

1000	900	800	700	600	550	500	450	N
30.5	27.4	24.3	21.23	20.9	19.7	19.5	19.3	t(ms) non indexing
10.34	9.2	8.62	10.85	8.96	9.32	8.72	10.4	t(ms) indexing

يبين الشكل (6-5) الفرق بين أزمنة البحث قبل الفهرسة وأزمنة البحث بعد الفهرسة:



الشكل (6-5) مقارنة النتائج قبل وبعد عملية الفهرسة

من خلال المقارنة بين المخططين الناتجين نلاحظ انخفاض زمن البحث، وهذا ما يلاحظ من بداية المخطط أي من أجل قيم صغيرة لعدد الملفات، وبالتالي الحد الأدنى لزمن البحث المستهلك بدون فهرسة يختلف بفارق ملحوظ عن الزمن المستهلك مع عملية الفهرسة، وهذا الفارق يزداد بازدياد عدد الملفات بشكل كبير وهنا تكمن أهمية الفهرسة.

الاستنتاجات التوصيات:

يمكننا القول إن هذا البحث قد حقق مجموعة من النتائج الجيدة يمكن أن نلخصها بالآتي:

1 -تخفيض أزمنة الوصول إلى ملفات الصور باستخدام الأرشفة.

2 -تقليل أزمنة البحث عن ملفات الصور بحيث تصبح أقل ما يمكن باستخدام عملية الفهرسة.

3 -ترتيب الملفات ذات التواجد العشوائي ضمن قاعدة بيانات بحيث تصبح أوساط التخزين ذو قدرة تخزينية عالية.

4 -يوصي البحث بتوسيع عملية فهرسة ملفات الصور باستخدام معالجة الصورة والمنطق الضبابي وتحقيق عمليات التخزين والاستعلام من خلال الشبكات العصبونية أو من خلال الخوارزميات الجينية.

5 - ويوصي أيضاً باستخدام تقنية الفهرسة عن طريق تصميم خوارزمية جديدة لإيجاد صفحات الويب الجديدة وفهرستها وإدخالها داخل قاعدة بيانات محرك البحث.

المراجع:

1- BENG, C. O; GAO, H; JINBAO, W. "*Indexing multi-dimensional data in a cloud system*". 1st ed, National University of Singapore, China, 2010,112.

2- TSENG, H. C. *"Internet Applications with Fuzzy Logic and Neural Networks*". Journal of engineering computing and Architecture U. S. A. Vol. 1, N. 2, 2007, 137-156.

3- BHANU, R. B. *"Uncertain spatial data handling: Modeling, indexing and query"*. Computers & Geosciences, U. S. A. Vol. 8, N. 1, 2007, 42-61

4- TANIAR, D; RAHAYU, j. W. "*Global B^+ tree indexin in parallel Database system*". 1 st ed, Monash University, Australia ,2014, 701-708.

5- ELMASRI, R. j; NAVATHE, S. B. "*Fundamentals of Database System*". 6th, University of Texas, United State, Sep. 2010, 1201.

6- CHRISTOPHER, D. M; RAGHAVAN, P; SCHÜTZE, H. "*Introduction to Information Retrieval*". University of Cambridge, England, 2009, 581.

7- ZHANG, J; ZHONG, R. J. "*Enhancing the B^+ tree by dynamic node popularity caching*". Information Processing Letters, U. S. A. Vol. 110, N. 7, 2010, 268–273.

8- BARRANCOA, C. D; CAMPAAB, J. R; MEDINAB, J. M. "*A B^+tree based indexing technique for fuzzy numerical data*", Department of Computer Science and Artificial Intelligence, University of Granada, Spain, 2008, 1449.

9- SILBERSCHATZ, F. H; SUDARSHAN, K. S. "*Database System Concepts and B^+ trees*". 5th Ed ,Cambridge University, London, 2011, 169.

10- SILBERSCHATZ, A; KORTH, H. "*Database System Concepts*". 6th ed , Lehigh University ,New York, 2013, 1376.

11- TROELSEN, A. "*Pro C# 2008 and the .net 3.5 platform*". 4th ed, Minnesota University, United States, 2008, 1370.