Performance Evaluation of GlusterFS and NFS in Cloud Environment

Dr. Ahmad Mahmoud Ahmad*

(Received 29 / 1 / 2024. Accepted 21 / 5 / 2024)

□ ABSTRACT □

In distributed storage, data is transmitted and stored remotely on the Cloud Storage Systems where data can be backed up and maintained and made available for the users anywhere. Outside users only see the Distributed File System (DFS) as a single storage device. Nowadays, the Need of storing huge amounts of data has become a great challenge. A DFS provides many advantages over a local file system such as reliability, scalability, security… etc. Therefore, it was necessary to improve storage techniques to achieve the desired goals by relying on distributing data on clusters of servers with high specifications and loaded with high-performance systems.

Although, there are a big set of distributed file systems, we have, in this paper, tested, from a performance point of view, two of the main known storage solution: GlusterFS, and NFS. As a distributed architecture, we studied their performance with different topologies and different transmitted data chunks, regarding: Read, Re-Read, Write, Re-Write Processes. Results showed that GlusterFS outperforms NFS, as NFS can only do a small subset of what GlusterFS offers.

Keywords: Distributed Storage, Cloud Storage, Cloud-Computing, Distributed File System DFS, GlusterFS, NFS, Hadoop

Copyright **:**Tishreen University journal-Syria, The authors retain the copyright under a CC BY-NC-SA 04

* **Techeaning Staff Member- Lecturer – Department of Computer Systems and Networks – Faculty of Informatics Engineering – Tishreen University – Lattakia- Syria. Email: ahmad.m.ahmad@tishreen.edu.sy**

تقييم أداء نظامي التخزين الموزع GlusterFS و NFS في البيئة السحابية

د. أحمد محمود أحمد*

(تاريخ الإيداع 29 / 1 / 2024. قُبِل للنشر في 21/ 5 / 2024)

□ ملخّص □

في التخزين الموزع، يتم نقل البيانات وتخزينها عن بعد على أنظمة التخزين السحابية، حيث يمكن نسخ البيانات احتياطيًا وصيانتها وإتاحتها للمستخدمين في أي مكان، يرى المستخدمون الخارجيون نظام الملفات الموزعة فقط كجهاز تخزين واحد وهو ليس سوى واجهة. في الوقت الحاضر، أصبحت الحاجة إلى تخزين كميات هائلة من البيانات تحديًا كبيرًا . يوفر نظام الملفات الموزعة (DFS) العديد من المزايا مقارنة بنظام الملفات المحلي مثل الموثوقية وقابلية التوسع والأمان وما إلى ذلك. لذلك كان من الضروري تحسين تقنيات التخزين لتحقيق الأهداف المرجوة من خلال الاعتماد على توزيع البيانات على مجموعات من الخوادم ذات كفاءة عالية المواصفات ومحملة بأنظمة عالية الأداء .على الرغم من وجود مجموعة كبيرة من أنظمة الملفات الموزعة، فقد قمنا في هذه الورقة البحثية باختبار اثنين من حلول التخزين الرئيسية المعروفة، GlusterFS و NFS، و مقارنة أدائهما، باعتبارهما بنية موزعة، قمنا بدراسة الأداء في بنى مختلفة و بحجوم معطيات متغيرة فيما يخص عمليات القراءة وإعادة القراءة والكتابة وإعادة الكتابة .أظهرت النتائج أن GlusterFSيتفوق علىNFS ، حيث أن NFS يمكنه فقط القيام بمجموعة فرعية صغيرة مما يقدمه GlusterFS.

الكلمات المفتاحية: التخزين الموزع، التخزين السحابي، الحوسبة السحابية، DFS نظام الملفات الموزعة، GlusterFS، NFS، Hadoop

حقوق النشر : مجلة جامعة تشرين- سورية، يحتفظ المؤلفون بحقوق النشر بموجب الترخيص CC BY-NC-SA 04

* **عضو هيئة تدريس – قسم النظم الشبكات الحاسوبية – كلية الهندسة المعلوماتية – جامعة تشرين – اللاذقية– سورية.**
Email: ahmad.m.ahmad@tishreen.edu.sy

مقدمة:

يوجد في يومنا هذا العديد من الشركات التي تعمل على تخزين بيانات مستخدميها بشكل موزع بالاعتماد على عدة أنظمة تخزين. تواجه هذه الشركات، مع ازدياد عدد المستخدمين وزيادة حجم البيانات الخاصة بهم، مشكلة كبيرة تتعلق بتأمين المساحة اللازمة. لذلك كان لابد من تحسين تقنيات التخزين بحيث تحقق الأهداف المرجوَة وذلك بالاعتماد على توزيع البيانات على عناقيد من الخوادم ذات المواصفات العالية والمحمل عليها نظم عالية الأداء.

يعد نظام الملفات مكونًا أساسيًا للنظام الحاسوبي، الذي يوفر وصولاً وإدارة متسقة لأجهزة التخزين .هناك بعض الاختلافات في نظم الملفات بين أنظمة التشغيل المختلفة، ولكن بشكل عام، توجد البيانات في شكل ملفات مزودة بواجهات APIs للوصول إليها مثل (Open, Read, Write, Seek and Close) إضافة الى تنظيم الملفات في دليل على شكل شجرة يوفر عمليات نقل الملفات أو الدلائل. تتصف معظم أنظمة الملفات بأنها ذاتية، وتوفر الوصول والإدارة لجهاز/أجهزة التخزين داخل نظام التشغيل، و لكن مع التطور السريع للإنترنت، تواجه أنظمة الملفات المستقلة العديد من التحديات كالسعة و الأداء، الموثوقية و التوافرية. نركز في هذا البحث على مفهوم الأداء.

يوجد العديد من أنظمة الملفات الموزعة و منها:

- **GlusterFS** [1] هو نظام ملفات شبكي موزع قابل للتوسع تم تطويره بواسطة شركة .Gluster Inc بالولايات المتحدة، مفتوح المصدر. ظهر الإصدار الأول منه في عام 2007 واستحوذت عليه لاحقاً شركة RedHat في عام 2011. و هو مناسب لمهام معالجة البيانات الضخمة ويمكن إجراء معالجة البيانات هذه على التخزين السحابي.
- **GFS** [2] هو نظام ملفات موزع من Google و يعد رائدًا وممثلًا نموذجيًا لأنظمة الملفات الموزعة، التي طورتها BigFiles في وقت مبكر.
- **Hadoop** [3] يعتمد نظام الملفات الموزعة Hadoop على نظام ملفات GFS Google. المكونات الرئيسية في HDFS هي NameNode التي تدير (Name Space) ومجموعة من DataNodes التي تخزن البيانات الفعلية في ملفات . يستخدم Hadoop (MapReduce) و هو أداة معالجة بيانات قابلة للتطوير ومتسامحة مع الأخطاء وتتيح معالجة كمية هائلة من البيانات بالتوازي مع العديد من عقد الحوسبة المنخفضة .
- **CephFS** [4] بدأ بدراسة أطروحة دكتوراه لـ Sage Weil لتنفيذ إدارة البيانات الوصفية الموزعة في عام 2012. تم الاستحواذ عليهمن قبل RedHat في عام 2014. تم تصميم Ceph ليكون نظام تخزين متسامحًا مع الأخطاء وقابلاً للتوسع حيث يلعب خادم بيانات التعريف (MDS) دورًا في حل هذه المشكلة. يعتمد Ceph على نموذج تخزين الكائنات، حيث يتم تخزين بيانات الملف بواسطة أجهزة تخزين الكائنات (OSDs) ويتم تخزين البيانات التعريفية بواسطة خوادم البيانات التعريفية (MDSs).
- **zFS** [5] تم تصميم zFS كنظام ملفات موزع يوفر قابلية التوسع الشاملة عن طريق فصل إدارة التخزين عن إدارة الملفات .تم بناؤه في عام 20014 بواسطة شركة Sun Microsystems وهو مجاني ومفتوح المصدر.
- **MooseFS** [6] هو نظام ملفات موزع تم تنفيذه بلغة الـ C، مفتوح المصدر، مستوحى من بنية GFS. وهو مناسب للاستخدام في سيناريوهات النسخ الاحتياطي للبيانات أو استرداد النسخ الاحتياطي.
- **NFS** [7] نظام ملفات الشبكة (NFS) هو بروتوكول شبكة لمشاركة الملفات الموزعة. يحدد نظام الملفات طريقة تخزين البيانات في شكل ملفات واسترجاعها من أجهزة التخزين، مثل محركات الأقراص الثابتة، ومحركات

الأقراص ذات الحالة الصلبة، ومحركات الأشرطة NFS .هو بروتوكول مشاركة ملفات الشبكة الذي يحدد طريقة تخزين الملفات واسترجاعها من أجهزة التخزين عبر الشبكات.

الجدير بالذكر أنه يوجد العديد من الأنظمة الأخرى [8] التي لم يتم ذكرها هنا مثل Tectonic ، JuiceFS، Lustre، Iozone ، مع العلم بأن تركيزنا سينصب على تظامي الملفات NFS و GlusterFS .

مشكلة البحث:

تكمن مشكلة البحث في مايلي:

1- اتجاه أنظمة الملفات إلى التوزع:

عند تخزين البيانات على خادم وحيد، ستصبح هذه البيانات عرضة للخسارة عند تعطل الخادم وخروجه عن الخدمة، لذلك كان لابد من اعتماد البنية الموزعة لضمان وجود أكثر من نسخة من البيانات بالإضافة لذلك الحصول على أداء أفضل عند توزيع الحمل على أكثر من خادم.

2- الحاجة إلى تلبية متطلبات الزبائن بالسرعة القصوى:

مع التقدم السريع في عالم الإنترنت، وزيادة وعي المستخدم للتكنولوجيا الحديثة بين يديه، أصبحت الشركات تتنافس على تقديم مستوى أفضل من التخديم، وذلك عبر تلبية طلبات المستخدمين بالسرعة القصوى لذلك كان لابد من اعتماد أفضل الأنظمة لتقديم الأداء المطلوب.

3- الحاجة لتحديد أفضلية نظام عن نظام آخر في ظل وجود أنظمة ملفات متنوعة وذلك عبر المقارنة بينها:

مع ظهور العديد من التقنيات الحديثة المعتمدة في تخزين البيانات، والتي يتم تحديثها بشكل دائم والمترافقة مع تطور المخدمات من الناحية العتادية، كان لابد من اجراء مقارنة لتحديد الأفضل بين هذه النظم والأكثرها توافقية مع نوع البيانات التي سيتم تخزينها.

أهمية البحث وأهدافه:

أهمية البحث:

تتلخص أهمية البحث في التركيز على مايلي:

1- الحاجة إلى تخزين بيانات كبيرة:

يظهر البحث تقييماً لأداء عدة نظم ملفات موزعة والذي يلقى رواجاً من قبل الجامعات والشركات لاسيما بعد التنامي السريع في عالم التواصل الاجتماعي الذي أدى بدوره إلى تزايد كبير في عدد المستخدمين لشبكة الإنترنت، بالتالي أصبحنا أمام مشكلة تضخم البيانات الخاصة بالمستخدمين مما أدى ذلك لظهور حاجة ملحة لتأمين بنية تحتية قادرة على تخزين هذه البيانات والتعامل معها من حيث السرعة في التخزين والسرعة في الاستعادة.

2- تسهيل عملية اختيار النظام المناسب لكل شركة:

وذلك من خلال تحديد الفروقات بين نظم التخزين الموزع واكتشاف ثغرات الأداء في كل نظام مع تحديد الخوادم الملائمة بالمواصفات المطلوبة.

أهداف البحث:

اكتشاف الفروقات بين أنظمة التخزين الموزع وتقييم أداءها من حيث البارامترات التالية:

1- الإنتاجية: كمية البيانات التي سيتم كتابتها أو قراءتها في الثانية.

2- التأخير: وذلك من خلال حساب الزمن المستغرق في عمليات الكتابة والقراءة.

3- حجم الـ chunks: والمقصود به حجم كتلة البيانات الافتراضي المعتمد من قبل كل نظام.

منهجية البحث:

1- دراسة نظم التخزين الموزع (NFS, GlusterFS) وآلية عملها في تخزين البيانات واسترجاعها.

2- اجراء التنفيذ العملي والمقارنات المطلوبة لتقييم الأداء بين النظم السابقة.

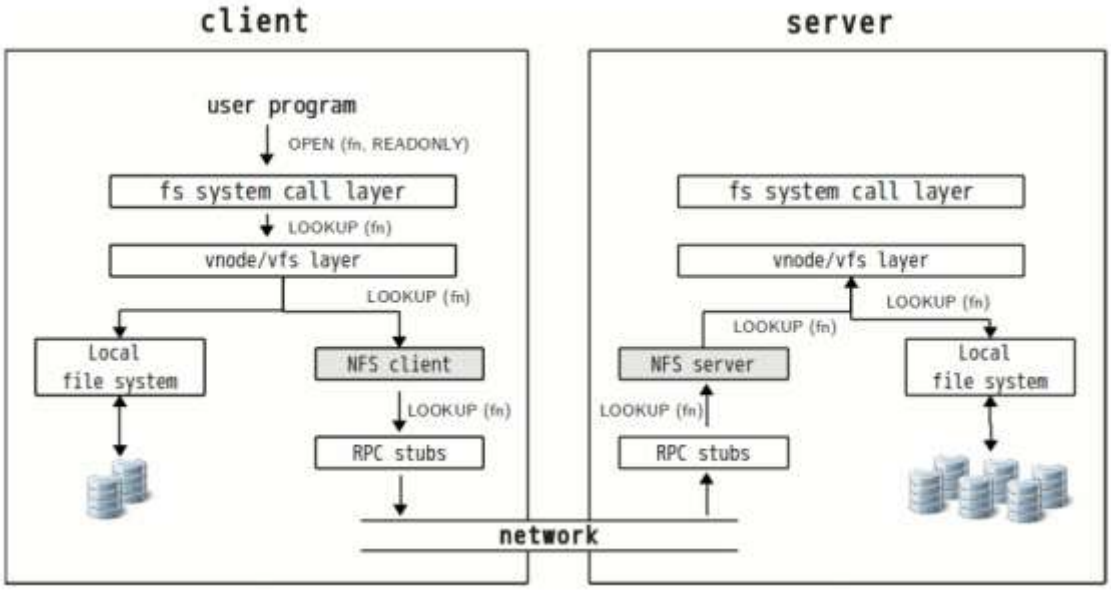
الأدوات المستخدمة:

- GlusterFS-client,GlusterFS-server
- nfs-kernel-server,nfs-common

أنظمة التخزين الموزع:

1- NFS (Network File System):

نظام تخزين موزع يسمح بإنشاء (mount points: مساحة تخزين مشتركة) على الخوادم مما يتيح إدارة مساحة التخزين في مواقع مختلفة والتحكم بعمليات االقراءة والكتابة إلى تلك المساحة من قبل العملاء (clients). يوفر NFS إمكانية الوصول إلى (mount points: مساحة تخزين مشتركة) بسرعة وسهولة وتعمل بشكل جيد في الحالات التي يجب الوصول إلى الموارد المشاركة عبر الشبكة. NFS هو معيار مفتوح المصدر ومعرف في Request for Comments (RFC 1813). يوضح الشكل (1) معمارية النظام NFS.



الشكل (1): مخطط يوضح معمارية النظام NFS.

2- GlusterFS:

هو نظام ملفات مجاني ومفتوح المصدر وقابل للتطوير، مصمم خصيصاً لتخزين كميات كبيرة من البيانات على سبيل المثال (التخزين السحابي) الذي يحتاج مساحة كبيرة للتخزين وإدارة عمليات تدفق البيانات من وإلى العميل (client). GlusterFS مكون من عنصرين الخادم والعميل، يقوم الخادم بتشغيل خدمة الـ GlusterFSD و يقوم العميل بإنشاء

mount pointsحيث يتعامل معها وكأنها مساحة تخزين محلية، يمكن زيادة الأداء من خلال وزيادة عدد العقد أو زيادة مساخة التخزين على العقد (الخوادم). GlusterFS مرخص وفق ترخيصGNU General Public License .

ميزات النظام GlusterFS:

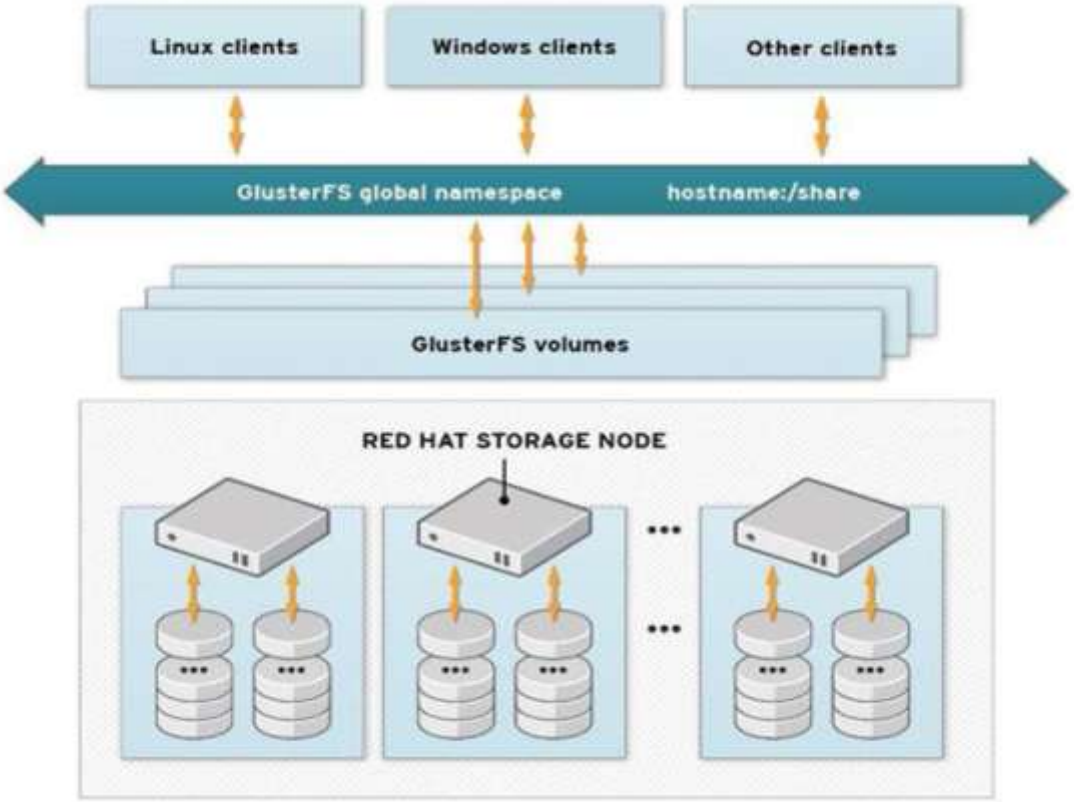
1- مساحة تخزين كبيرة.

2- قابلية التوسع.

3- تحقيق التوافرية حيث يوجد نسخ متماثلة من البيانات على جميع العقد.

4- التسامح مع الأخطاء وإعادة توازن البيانات في حال حدوث مشاكل.

يوضح الشكل (2) معمارية النظام GlusterFS.



الشكل (2): مخطط يوضح معمارية النظام GlusterFS.

الدراسات المرجعية:

تتلخص مشكلة البحث في [1] في تقييم أداء NFSv4 الذي أصبح حلاً ذو شعبية كبيرة في عالم التخزين عبر الشبكة، والذي يختلف إلى حد كبير عن الاصدارات السابقة له، حيث يركز البحث على المقارنة بين NFSv4 و NFSv3 وذلك لإظهار الميزات التي أتى بها NFSv4 من حيث التفويض للعميل بالاحتفاظ بنسخة مؤقتة من جانبه إضافة لميزة النسخ المتكرر. أظهرت النتائج تفوق NFSv4 على NFSv3 من حيث الإنتاجية في قراءة الملفات صغيرة الحجم.

يتناول البحث [2] الإصدار الأخير من NFS و هو NFSv4 والتي تزيد من أداء التخديم في تخزين البيانات وتأمين الوصول إليها بأسرع وقت. بينت النتائج أن تخفيض عدد حزم الـ IP التي يتم تبادلها عبر الشبكة مفيد أكثر لزيادة أداء الشبكة.

أما في البحث [3]، فقد بدأت المشكلة عند حدوث مشكلة عنق الزجاجة لدى الخادم وذلك عند نفاذ الذاكرة العشوائية، والذي يحدث نتيجة لعمليات الدخل والخرج (القراءة والكتابة الى القرص الصلب). للعلم فإن الـ GlusterFS يعمل على مستوى عالي من التخزين حيث يستخدم ال scale out وخوارزمية elastic hash لحل اختناقات الشبكة الناتجة عن عمليات الدخل والخرج المتزايدة إلى القرص الصلب، لكن مع زيادة عدد الخوادم يمكن تحسين الأداء بشكل أفضل وحل اختناقات التخزين بشكل أسرع. بينت النتائج أن زيادة الأداء مع زيادة عدد الخوادم.

عرض البحث [4] المشكلة والتي تتلخص في الحاجة إلى تخزين كمية كبيرة من البيانات بحيث يتم تخزينها بشكل آمن وأكثر فاعلية من ناحية الأداء وذلك بعد التطور السريع التطبيقات المستخدمة عبر الإنترنت وتنامي البنية التحتية للحوسبة السحابية وفقاً لذلك. بينت النتائج أن GlusterFS يقدم أداء أقل كفاءة عند إضافة وقراءة ملفات بدون تشفير.

من جهة أخرى، فقد تطرق البحث [5]، الى المشكلة في قياس أداء النظامHDFS (Hadoop Distributed File System) مع عمليات القراءة والكتابة للملفات الصغيرة والكبيرة الحجم والذي من الممكن أن يعطينا نتائج تحسن من آلية عمل نظام. HDFS . بينت النتائج أن أداء HDFS سيء عندما يكون حجم الملف أقل من حجم الكتلة الافتراضي في حين أنه أفضل عندما يكون حجم الملف أكبر من حجم الكتلة الافتراضي.

أما في البحث [6]، فتتلخص مشكلة البحث في تحسين أداء الـ Hadoop مع الملفات صغيرة الحجم وذلك في ظل التضخم السريع في حجم البيانات، فكان لابد من تحسين البنية التحتية للـ Hadoop في وجه مشكلة الاستهلاك الكبير للذاكرة بالإضافة لمشكلة عنق الزجاجة عند NameNode عند طلب عدد كبير من الملفات صغيرة الحجم. لوحظ تحسن أداء ال Hadoop بشكل كبير عند استخدام الحل HAR.

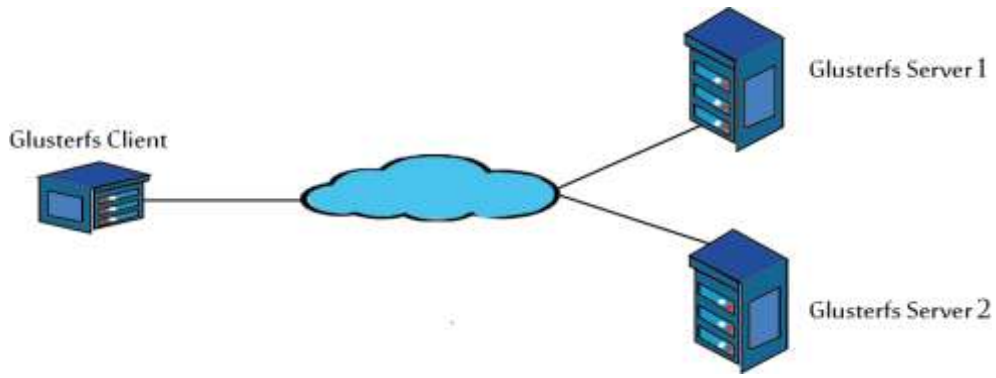
ستتركز دراستنا في هذه الورقة البحثية على اجراء المقارنة بين GlusterFS و NFS.

الجزء العملي والتجارب:

أستناداً الى الدراسات المرجعية، يركز الجزء العملي على اختبار بنيتين أساسيتين وهما البيئة GlusterFS و البيئة NFS و ذلك لأهميتهما الكبيرة إضافةً الى التركيز الكبير على تلك البنيتين من قبل الباحثين.

1- تجربة بيئة GlusterFS:

تم بداية تجهيز البيئة GlusterFS وفق الشكل (4). قمنا باختيار مخدمين اثنين GlusterFS مع زبون GlusterFS، و جميعها تتوضع في أمستردام عاصمة هولندا.



الشكل (4): مخطط يوضح البنية المستخدمة GlusterFS

يوضح الجدول (1) مواصفات المخدمات GlusterFS والزبون المستخدمة في التجربة.

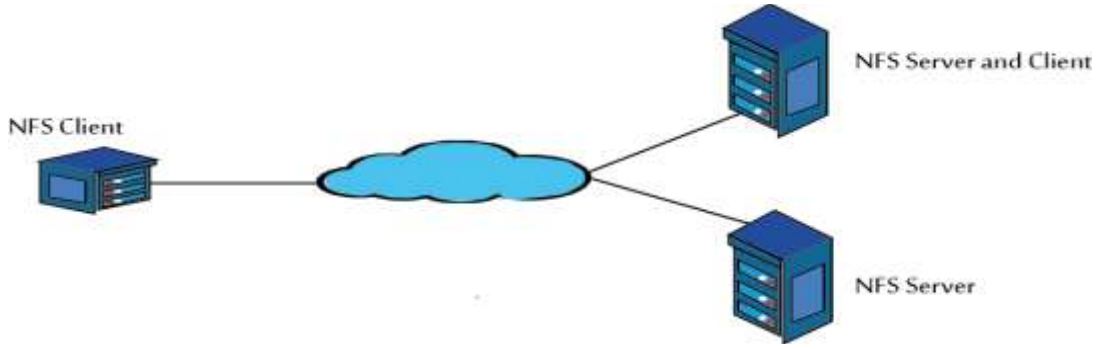
الجدول (1): مواصفات المخدمات GlusterFS المستخدمة

	Location	IP	CPU	RAM	System	kernel	NFS version
GlusterFS server 1	Netherlands Amsterdam	167.99.217.243	Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz * 6	16 GB	Ubuntu 18.04.1	generic-4.15.0-62	GlusterFS 3.13.2
GlusterFS server 2	Netherlands Amsterdam	178.62.204.241	Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz * 6	8 GB	Ubuntu 18.04.3	generic-4.15.0-65	GlusterFS 3.13.2
GlusterFS client	Netherlands Amsterdam	128.199.47.114	Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz * 8	16 GB	Ubuntu 18.04.1	generic-4.15.0-55	GlusterFS 5.12

لتحقيق التجارب، قمنا بتجهيز بيئة العمل بدءا من لحظة الـ Setup وصولا لمرحلة الإعداد والتحضير للتجارب الممكنة. تم نقل بيانات بين السيرفرات بالأحجام التالية: (10 ميغابايت 100 ميغابايت، 1 غيغابايت، 10 غيغابايت). تم بعد ذلك تسجيل الزمن المستغرق ومراقبة الحمل على السيرفرات لمدة 15 يوم. تم الحصول على ملف log فيه الأزمنة المستغرقة لنقل البيانات خلال مدة التجريب.

2- تجربة بيئة NFS:

تم العمل على النقاط التالية بداية بتجهيز البيئة NFS وفق الشكل (5):



الشكل (5): مخطط يوضح البنية المستخدمة NFS

بينما يوضح الجدول (2) مواصفات المخدمات والزبون NFS.

الجدول (2): مواصفات المخدمات NFS المستخدمة

	Location	IP	CPU	RAM	System	kernel	NFS version
NFS server	Syrian Arab Republic - Damascus	185.216.135.11	Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz * 24	16 GB	Ubuntu 14.04.6	generic-3.19.0-25	3
NFS server and client	Netherlands Amsterdam	167.99.217.243	Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz * 6	16 GB	Ubuntu 18.04.1	generic-4.15.0-6	3
NFS client	Netherlands Amsterdam	128.199.47.114	Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz * 8	16 GB	Ubuntu 18.04.1	4.15.0-55-generic	3

كما الحال في التجربة GlusterFS، لتحقيق التجارب، قمنا بتجهيز بيئة العمل بدءاً من لحظة الـ setup وصولا لمرحلة الإعداد والتحضير للتجارب الممكنة. تم نقل بيانات بين السيرفرات بالأحجام التالية: (10 ميغابايت، 100 ميغابايت، 1 غيغابايت، 10 غيغابايت). تم بعد ذلك تسجيل الزمن المستغرق ومراقبة الحمل على السيرفرات لمدة 15 يوم.

النتائج والمناقشة:

شملت النتائج أربع عمليات أساسية وهي: عملية القراءة (Read Process)، عملية إعادة القراءة (Re-Read Process)، عملية الكتابة (Write Process)، عملية إعادة الكتابة (Re-Write Process)، وذلك لكلا البنتين السابقتين.

1-عملية القراءة (Read Process):

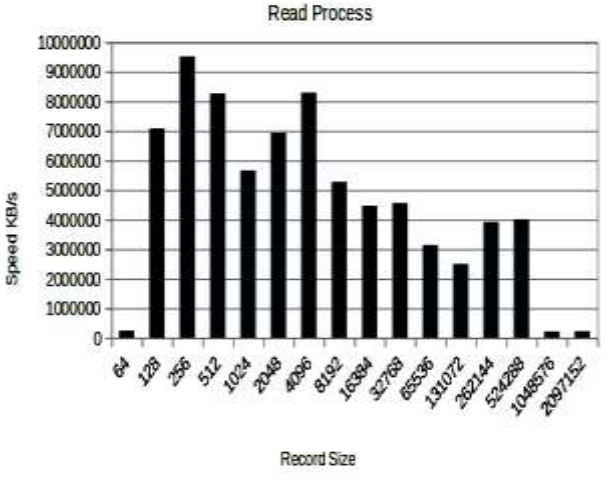
a) - البنية NFS:

نلاحظ أن أداء القراءة يختلف باختلاف حجم الملف حيث يبدأ بسرعة أقل من 1 ميغابايت بالثانية ثم يزداد تدريجياً حيث يصل لأعلى سرعة وهي 9 غيغابايت بالثانية وذلك عندما يصل حجم الملف لـ 256 كيلوبايت، يبدأ الأداء بالعودة لسرعة أقل من 1 ميغابايت بالثانية عندما يصل حجم الملف لـ 1 غيغابايت.

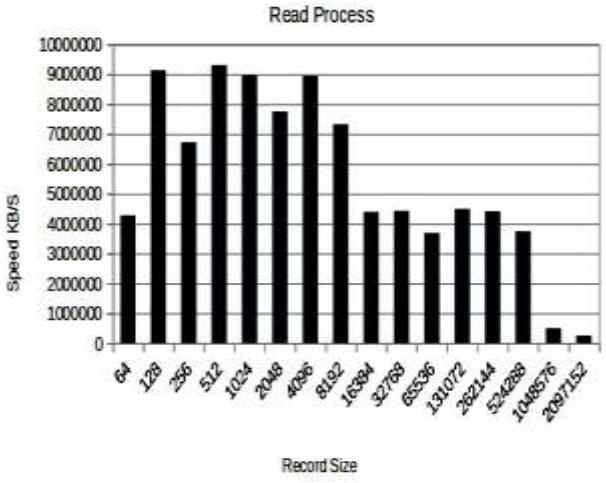
b) - البنية GlusterFS:

نلاحظ أن أداء القراءة يختلف باختلاف حجم الملف حيث يبدأ بسرعة أفضل من NFS مع الملفات صغيرة حجم ثم يزداد بسرعة حيث يصل لأعلى سرعة وهي 9 غيغابايت بالثانية وذلك عندما يصل حجم الملف لـ 128 كيلوبايت، يبدأ الأداء بالعودة لسرعة أقل من 1 ميغابايت بالثانية عندما يصل حجم الملف لـ 1 غيغابايت.

من الملاحظ أن أداء GlusterFS أفضل من أداء NFS مع الملفات الصغيرة بالنسبة لعملية القراءة، بينما يعتبر الأداء متقارب جداً مع زيادة حجم الملف.



NFS



GlusterFS

2-عملية إعادة القراءة (Re-Read Process):

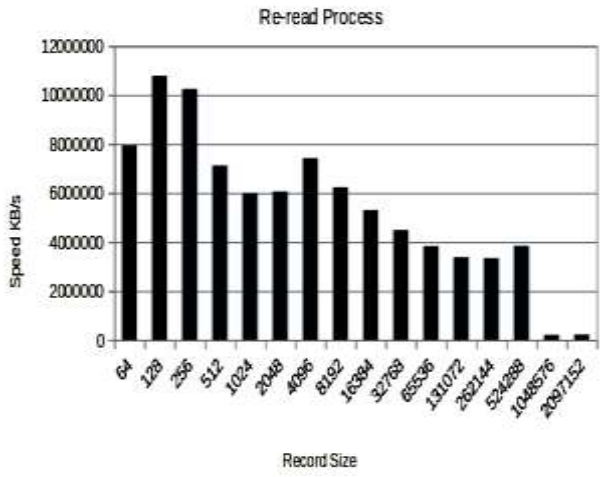
a) - البنية NFS:

نلاحظ أن أداء عملية إعادة القراءة يختلف باختلاف حجم الملف حيث يبدأ بسرعة أفضل بكثير منها في عملية القراءة ثم يزداد ويصل لأعلى سرعة وهي بين 12 – 10 غيغابايت بالثانية وذلك عندما يصل حجم الملف ل 128 كيلوبايت، يبدأ الأداء بالعودة لسرعة أقل من 1 ميغابايت بالثانية عندما يصل حجم الملف لـ 1 غيغابايت.

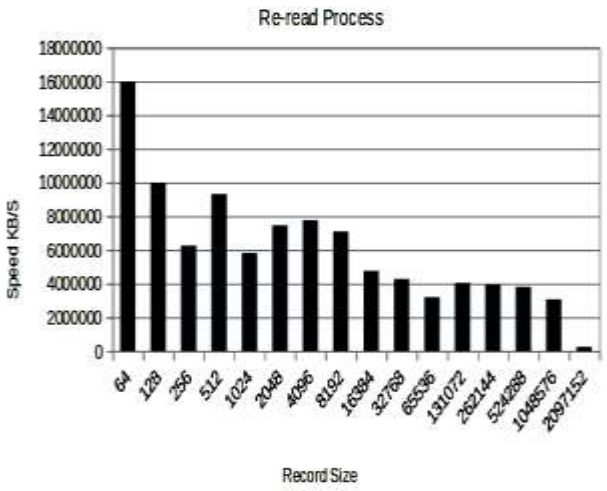
b) - البنية GlusterFS:

نلاحظ أن أداء عملية إعادة القراءة يختلف باختلاف حجم الملف حيث يبدأ أيضاً بسرعة أفضل من NFS مع الملفات صغيرة الحجم لكن ينخفض بسرعة ويصل لسرعة أقل من 1 ميغابايت بالثانية عندما يصل حجم الملف لـ 2 غيغابايت.

من الملاحظ أن أداء GlusterFS أقل من أداء NFS بالمجمل بالنسبة لعملية إعادة القراءة.



NFS



GlusterFS

3-عملية الكتابة (Write Process):

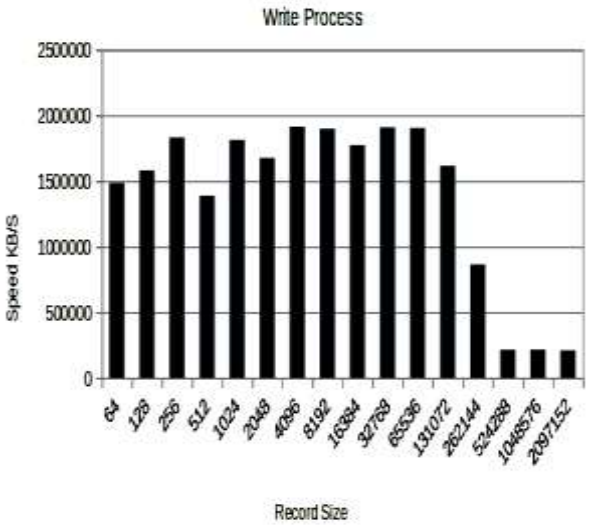
a) - البنية NFS:

نلاحظ أن أداء الكتابة يختلف باختلاف حجم الملف حيث يبدأ بسرعة تتراوح من 1 – 1.5 غيغابايت بالثانية إلى أن يتجاوز حجم الملف 512 ميغابايت حيث ثم ينخفض الأداء بالعودة لسرعة أقل من 5 ميغابايت بالثانية.

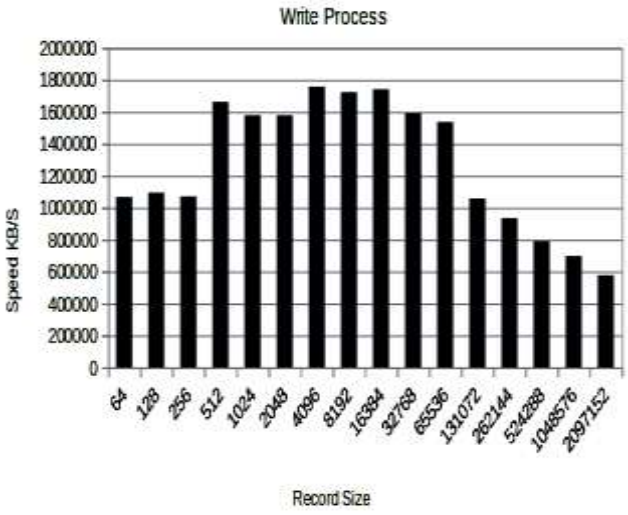
b) - البنية GlusterFS:

نلاحظ أن أداء الكتابة يختلف باختلاف حجم الملف حيث يبدأ بسرعة أقل من NFS مع الملفات صغيرة الحجم ثم يزداد حتى يصل لسرعة متقاربة من السرعة التي تقدمها NFS لكن مع زيادة حجم الملف فإنه يحافظ على سرعة أفضل من NFS، و أن أداء GlusterFS أقل من أداء NFS بالمجمل بالنسبة لعملية إعادة القراءة،

من الملاحظ أن أداء GlusterFS أفضل من أداء NFS مع الملفات كبيرة الحجم بالنسبة لعملية الكتابة حيث يحافظ على سرعة أفضل نسبياً من السرعة التي تقدمها NFS.



NFS



GlusterFS

3-عملية إعادة الكتابة (Re-Write Process):

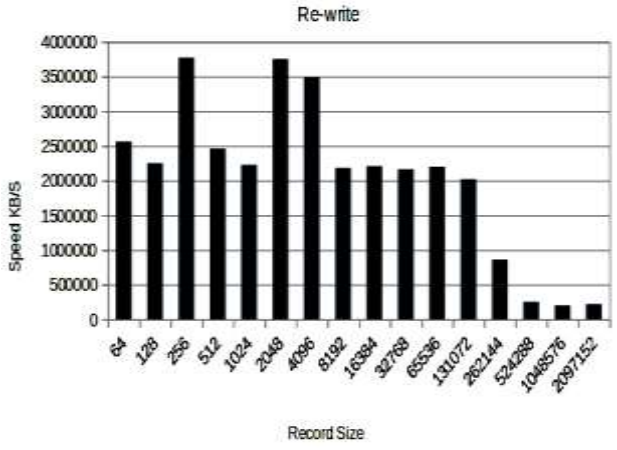
a) - البنية NFS:

نلاحظ أن أداء إعادة الكتابة يختلف باختلاف حجم الملف حيث يبدأ بسرعة تتجاوز 2.5 غيغابايت بالثانية وحين يتجاوز حجم الملف 512 ميغابايت ينخفض الأداء بالعودة لسرعة أقل من 5 ميغابايت بالثانية.

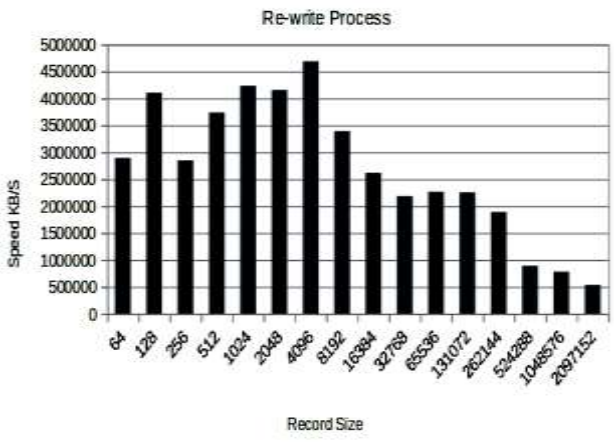
b) - البنية GlusterFS:

نلاحظ أن أداء إعادة الكتابة يختلف باختلاف حجم الملف حيث يبدأ بسرعة متقاربة من السرعة التي تقدمها NFSمع الملفات صغيرة الحجم ثم يزداد حتى يصل لسرعة تتجاوز 4.5 غيغابايت بالثانية ثم ينخفض الأداء مع زيادة حجم الملف لكن يبقى محافظاً على سرعة أفضل من NFS.

من الملاحظ أن أداء GlusterFS أفضل من أداء NFS مع الملفات كبيرة الحجم بالنسبة لعملية إعادة الكتابة حيث يحافظ على سرعة أفضل نسبياً من السرعة التي تقدمها NFS.



NFS



GlusterFS

الاستنتاجات والتوصيات:

قمنا في هذا البحث بمقارنة أداء النظام NFS مع النظام GlusterFS وذلك من خلال أربع عمليات وهي: عملية القراءة (Read Process)، عملية إعادة القراءة (Re-Read Process)، عملية الكتابة (Write Process)، عملية إعادة الكتابة (Re-Write Process). من الملاحظ أن أداء GlusterFS أفضل من أداء NFS مع الملفات الصغيرة بالنسبة لعملية القراءة. كما لوحظ أن أداء GlusterFS أفضل من أداء NFS مع الملفات كبيرة الحجم بالنسبة لعملية الكتابة، و أن أداء GlusterFS أفضل من أداء NFS مع الملفات كبيرة الحجم بالنسبة لعملية إعادة الكتابة. بالمجمل، تبين النتائج تفوق GlusterFS على NFS، حيث يمكن لـ NFS فقط القيام بمجموعة فرعية صغيرة مما يقدمه GlusterFS. يمكننا مستقبلاً زيادة عدد المخدمات، عدد العملاء و عدد بطاقات الشبكة للحصول على نتائج أكثر. كما يمكننا اجراء مقارنات مع أنظمة تخزين أخرى مثل: Hadoop،Iozone، Ceph….الخ.

References:

1. Rodrigo Leite;Priscila Solis, "Performance Analysis of Data Storage in a Hyperconverged Infrastructure Using Docker and GlusterFS", 2019 XLV Latin American Computing Conference (CLEI), IEEE 2019.
2. Chitresh Verma;Rajiv Pandey, "Comparative Analysis of GFS and HDFS: Technology and Architectural landscape", 2018 10th International Conference on Computational Intelligence and Communication Networks (CICN), IEEE 2018.
3. Manish Kumar Gupta;Shrawan Kumar Pandey;Anish Gupta, "HADOOP- An Open Source Framework for Big Data", 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), IEEE 2022.
4. Kisik Jeong;Carl Duffy;Jin-Soo Kim;Joonwon Lee, "Optimizing the Ceph Distributed File System for High Performance Computing" 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2019.
5. Tinnaphob Angkaprasert;Kasidit Chanchio, "A Backup Mechanism of Virtual Machine Checkpoint Image using ZFS Snapshots", 2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE), IEEE 2023.
6. Yucheng Fang;Huibiao Zhu;Gang Lu;Lili Xiao;Wanling Xie, "Modeling and Verifying MooseFS in CSP" 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), IEEE 2018.
7. Luca Acquaviva;Paolo Bellavista;Antonio Corradi;Luca Foschini;Leo Gioia;Pasquale Carlo Maiorano Picone, "Cloud Distributed File Systems: A Benchmark of HDFS, Ceph, GlusterFS, and XtremeFS", 2018 IEEE Global Communications Conference (GLOBECOM) IEEE 2018.
8. Dan Sullivan, "Designing Storage Systems",Official Google Cloud Certified Professional Cloud Architect Study Guide, Year: 2020 | Book Chapter | Publisher: Wiley.
9. Ming Chen, Dean Hildebrand, Geoff Kuenning, "Newer Is Sometimes Better: An Evaluation of NFSv4.1" Stony Brook University; IBM Research—Almaden; Harvey Mudd College, 2015.
10. Ghania Al Sadi "Tuning and Optimizing Network File System Server Performance" International Journal of Computer Applications (0975 – 8887) Volume 134 – No.10, January 2016.

11. Dawei XIAO,Cheng ZHANG,Xiaodong LI "The Performance Analysis of GlusterFS In Virtual Storage", International Conference on Advances in Mechanical Engineering and Industrial Informatics (AMEII 2015).

12. Roopali VIJ "Performance Measurements And Comparison For Gluster FS And Azure Blob Storage" International Journal Of Scientific & Technology Research Volume 4, Issue 12, December 2015.

13. T Lakshmi Siva Rama Krishna,Dr T Ragunathan,Sudheer Kumar Battula "Performance Evaluation of Read and Write Operations in Hadoop Distributed File System"2014 Sixth International Symposium on Parallel Architectures, Algorithms and Programming.

14. Awais Mehmood,Waqas Mehmood,Muhammad Usman,Yasmeen Khaliq "Performance Efficiency in Hadoop for Storing and Accessing Small Files"The Seventh International Conference on Innovative Computing Technology (INTECH 2017).