

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION**

VALTRUS INNOVATIONS LTD. and
KEY PATENT INNOVATIONS LTD.,

Plaintiffs,

V.

NETAPP, INC.,

Defendant.

Case No. 2:25-cv-

JURY TRIAL DEMANDED

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiffs Valtrus Innovations Ltd. (“Valtrus”) and Key Patent Innovations Ltd. (“KPI”) (collectively, “Plaintiffs”) for their Complaint against Defendant NetApp, Inc. (“NetApp” or “Defendant”) for patent infringement allege as follows:

THE PARTIES

1. Valtrus is the successor-in-interest to a substantial patent portfolio created by Hewlett Packard Enterprise and its predecessor, subsidiary, and affiliate companies, including Hewlett-Packard Development Company, L.P. (collectively, “HPE”). Valtrus is an Irish entity duly organized and existing under the laws of Ireland. The address of the registered office of Valtrus is: The Glasshouses GH2, 92 Georges Street Lower, Dun Laoghaire, Dublin A96 VR66, Ireland. HPE’s worldwide corporate headquarters is located in Houston, Texas. One of HPE’s primary U.S. facilities is located in Plano, Texas.

2. KPI is the beneficiary of a trust pursuant to which Valtrus owns, holds, and asserts the Patents-in-Suit as defined in Paragraph 18. KPI is an Irish entity duly organized and existing

under the laws of Ireland. The address of the registered office of KPI is: The Glasshouses GH2, 92 Georges Street Lower, Dun Laoghaire, Dublin A96 VR66, Ireland.

3. On information and belief, NetApp is a company organized under the laws of the State of Delaware, with a place of business in this District located at 6900 Dallas Pkwy Suite #110, Plano, TX 75024. NetApp may be served through its registered agent for service, Corporation Service Company located at 251 Little Falls Drive, Wilmington, Delaware 19808. On information and belief, NetApp does business in Texas, directly or through intermediaries, and offers its products and/or services, including those accused herein of infringement, to customers and potential customers located in Texas, including in the Judicial District of the Eastern District of Texas.

4. NetApp controls, participates in the commission of, and has a direct financial interest in the infringing acts set forth herein.

JURISDICTION AND VENUE

5. This is an action for patent infringement arising under the patent laws of the United States, 35 U.S.C. §§ 1, *et seq.* This Court has jurisdiction over this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

6. This Court has personal jurisdiction over Defendant. Defendant regularly conducts business and has committed acts of patent infringement and/or have induced acts of patent infringement by others in this Judicial District and/or has contributed to patent infringement by others in this Judicial District, the State of Texas, and elsewhere in the United States. Defendant, directly and/or through subsidiaries or intermediaries, has committed and continues to commit acts of infringement in this District by, among other things, making, using, importing, offering to sell, and/or selling products that infringe the Patents-in-Suit as defined in Paragraph 16.

7. This Court has general jurisdiction over NetApp due to its continuous and systematic contacts with the State of Texas, including through its operation of physical offices in the State of Texas, including at 6900 Dallas Pkwy Suite #110, Plano, TX 75024, its targeting of Texas residents with the <https://www.netapp.com/> website, its ownership and/or lease of land in the State of Texas, and other business activities throughout the State of Texas.

8. NetApp is subject to the specific personal jurisdiction of this Court because Plaintiffs' patent infringement claims against NetApp specifically arise from NetApp's acts of infringement in the State of Texas. These acts of infringement include directing and controlling the operation of the <https://www.netapp.com/> website, using the patented inventions, and specifically targeting residents of Texas with this website to further the sale of products and services to those customers online in this District.

9. On information and belief, NetApp operates and/or contracts with third parties for manufacturing and supply chain solutions, including in the State of Texas.

Manufacturing and Supply Chain

We have outsourced manufacturing operations to third parties located in Fremont, California; San Jose, California; San Antonio, Texas; Guadalajara, Mexico; Schiphol Airport, The Netherlands; Tiszaujvaros, Hungary; Wuxi, China; Taoyuan City, Taiwan; and Singapore. These operations include materials procurement, commodity management, component engineering, test engineering, manufacturing engineering, product assembly, product assurance, quality control, final test, and global logistics. We rely on a limited number of suppliers for materials, as well as several key subcontractors for the production of certain subassemblies and finished

¹

10. Venue is proper in this District under 28 U.S.C. §§ 1391(b)-(d) and 1400(b). Defendant is registered to do business in the State of Texas, has offices in the State of Texas, has transacted and continues to transact business in the Eastern District of Texas, and has committed and continues to commit acts of direct and indirect infringement in the Eastern District of Texas.

11. Defendant is subject to this Court's jurisdiction pursuant to due process and/or the Texas Long Arm Statute due at least to its substantial business in this State and Judicial District,

¹ NetApp Form 10-K, filed with the SEC on June 10, 2024, at pg. 9, available at: <https://investors.netapp.com/static-files/16e8aaf6-cff6-4d38-991c-d92972629339>

including (a) at least part of its past infringing activities, (b) regularly doing or soliciting business in Texas, and/or (c) engaging in persistent conduct and/or deriving substantial revenue from goods and services provided to customers in Texas.

PATENTS-IN-SUIT

12. On February 5, 2013, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 8,370,416 (the “’416 Patent”) entitled “Compatibility Enforcement in Clustered Computing Systems.” A true and correct copy of the ’416 Patent is attached hereto as Exhibit A.

13. On February 5, 2013, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 8,370,571 (the “’571 Patent”) entitled “Transfer Control of a Storage Volume Between Storage Controllers in a Cluster.” A true and correct copy of the ’571 Patent is attached hereto as Exhibit B.

14. On March 8, 2011, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 7,904,686 (the “’686 Patent”) entitled “Data Security for Use with a File System.” A true and correct copy of the ’686 Patent is attached hereto as Exhibit C.

15. On December 21, 2010, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 7,856,488 (the “’488 Patent”) entitled “Electronic Device Profile Migration.” A true and correct copy of the ’488 Patent is attached hereto as Exhibit D.

16. On October 10, 2006, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 7,120,832 (the “’832 Patent”) entitled “Storage Device Performance Monitor”. A true and correct copy of the ’832 Patent is attached hereto as Exhibit E.

17. On December 29, 2009, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 7,640,332 (the “’332 Patent”) entitled “System and Method for Hot

Deployment/Redeployment in Grid Computing Environment”. A true and correct copy of the ’332 Patent is attached hereto as Exhibit F.

18. Plaintiffs are the sole and exclusive owner of all legal right, title, and interest in the ’416 Patent, the ’571 Patent, the ’686 Patent, the ’488 Patent, the ’832 Patent, and the ’332 Patent (collectively, the “Patents-in-Suit” or “Asserted Patents”), and hold the exclusive right to take all actions necessary to enforce their rights to the Patents-in-Suit, including the filing of this patent infringement lawsuit. Plaintiffs also have the right to recover all damages for past, present, and future infringement of the Patents-in-Suit and to seek injunctive relief as appropriate under the law.

FACTUAL ALLEGATIONS

19. The ’416 Patent and the ’488 Patent generally relate to a system and method for a network comprising electronic devices associated with user profiles to allow migration of the user profiles from one electronic device to another. The technology described in the ’416 Patent was developed by David J. Hoover, Bernard A. Reed, III, Marjorie Krueger, and Douglas L. Voigt at Hewlett-Packard Development Company, L.P. The technology described in the ’488 Patent was developed by Daniel N. Cripe, David Kasperperson, and Charles L. Hudson at Hewlett-Packard Development Company, L.P. For example, the technology is implemented in NetApp’s Cluster Management ONTAP.

20. The ’571 Patent generally relates to a system and method for a file storage controller that is part of a cluster of storage controllers and includes an interface to communicate with at least one storage subsystem having a storage volume, where the first storage controller is initially assigned to process requests for the storage volume. The technology described in the ’571 Patent was developed by Daniel J. Mazina at Hewlett-Packard Development Company, L.P. For

example, the technology is implemented in NetApp SANtricity on E-Series controllers.

21. The '686 Patent generally relates to a system and apparatus for providing data security for use with a file system. The technology described in the '686 Patent was developed by Ranganath G. Iyengar at Hewlett-Packard Development Company, L.P. For example, the technology is implemented by NetApp's use of Apache Software Foundation's Hadoop Distributed File System ("HDFS").

22. The '832 Patent generally relates to technology for monitoring performance of a storage device by intercepting communications between a computer system and said storage device, analyzing the intercepted communications, and reallocating at least some of said data on said storage device to enhance the performance of the storage device. The technology described in the '832 Patent was developed by Kevin Collins and Michael P. Fleischmann at Hewlett-Packard Development Company, L.P. For example, the technology is implemented in systems using NetApp's SANtricity software as well as in NetApp's systems that implement Apache Kafka Streams using RocksDB Universal Style Compaction, RocksDB Universal Style Compaction, Apache Cassandra, and/or Apache HBase.

23. The '332 Patent generally relates to, e.g., steps for hot deployment and/or redeployment in a grid computing environment having one or more grid nodes, wherein the steps comprise adding a new version of an application from a repository server, determining which of the one or more grid nodes are running the application affected by the new version of the software, notifying the client application manager associated with one or more of the affected grid nodes, and hot deploying/redeploying the new version of the application to the affected grid nodes. The technology described in the '332 Patent was developed by Sanjay Dahiya at Hewlett-Packard Development Company, L.P. For example, the technology is used by NetApp and/or is

implemented in systems, such as NetApp ONTAP and NetApp Trident using Kubernetes for rolling updates.

24. NetApp has infringed the '832 Patent by making, using, selling, offering to sell, and/or importing, and by actively inducing others to make, use, sell, offer to sell, and/or import the infringing products identified above.

25. NetApp has infringed and is continuing to infringe the '416 Patent, the '571 Patent, the '686 Patent, the '488 Patent, and the '332 Patent by making, using, selling, offering to sell, and/or importing, by actively inducing others to make, use, sell, offer to sell, and/or import, and/or contributing to other's making, using, selling, offering to sell, and/or importing of the infringing products identified above.

COUNT I
(Infringement of the '416 Patent)

26. Paragraphs 1 through 25 are incorporated by reference as if fully set forth herein.

27. Plaintiffs have not licensed or otherwise authorized Defendant to make, use, offer for sale, sell, or import any products that embody the inventions of the '416 Patent.

28. Defendant has directly infringed and continues to directly infringe the '416 Patent, either literally or under the doctrine of equivalents, without authority and in violation of 35 U.S.C. § 271, by making, using, offering to sell, selling, and/or importing into the United States products that satisfy each and every limitation of one or more claims of the '416 Patent. Such products include, but are not limited to, NetApp's Cluster Management ONTAP.

29. For example, Defendant has directly infringed and continues to directly infringe at least claim 1 of the '416 Patent by making, using, offering to sell, selling, and/or importing into the United States products that include NetApp's Cluster Management ONTAP.

30. NetApp's Cluster Management ONTAP performs a method of creating a clustered computing system. For example, NetApp's ONTAP (operating system) uses either command line interface (CLI) or the Oncommand unified manager for creating and setting up a cluster.

Initialize and configure the cluster

You initialize the cluster by setting an administrative password for the cluster and setting up the cluster management and node management networks. You can also configure services like a domain name server (DNS) to resolve host names and an NTP server to synchronize time.

Steps

1. On a web browser, enter the node-management IP address that you have configured: "https://node-management-IP"
System Manager automatically discovers the remaining nodes in the cluster.
2. In the **Initialize Storage System** window, perform the following:
 - a. Enter cluster management network configuration data.
 - b. Enter Node management IP addresses for all the nodes.
 - c. Provide DNS details.
 - d. In the **Other** section, select the check box labeled **Use time service (NTP)** to add the time servers.

When you click **Submit**, wait for the cluster to be created and configured. Then, a validation process occurs.

What's Next?

After both clusters have been set up, initialized, and configured, perform the [Set up MetroCluster IP peering] procedure.

31. NetApp's Cluster Management ONTAP performs the step of storing license information for a computing cluster in a memory module associated with a computing node. For example, ONTAP store license information for a computing cluster either at the factory or adding them to the System Manager.

² <https://docs.netapp.com/us-en/ontap-metrocluster/install-ip/set-up-mcc-site-system-manager.html#other-systems>

You can install NetApp license files (NLFs) and license keys using System Manager, which is the preferred method for installing NLFs, or you can use the ONTAP CLI to install license keys. In ONTAP 9.10.1 and later, features are enabled with a NetApp license file, and in releases earlier than ONTAP 9.10.1, ONTAP features are enabled with license keys.

Steps

If you have already [downloaded NetApp license files](#) or license keys, you can use System Manager or the ONTAP CLI to install NLFs and 28-character license keys.

System Manager - ONTAP 9.8 and later System Manager - ONTAP 9.7 and earlier CLI

1. Select **Cluster > Settings**.
2. Under **Licenses**, select →.
3. Select **Browse**. Choose the NetApp License File you downloaded.
4. If you have license keys you want to add, select **Use 28-character license keys** and enter the keys.

³

You can use System Manager or the ONTAP CLI to view and manage licenses installed on your system, including viewing the license serial number, checking the status of a license, and removing a license.

View details about a license

Steps

How you view details about a license depends on what version of ONTAP you are using and whether you use System Manager or the ONTAP CLI.

System Manager - ONTAP 9.8 and later System Manager - ONTAP 9.7 and earlier CLI

1. To view details about a specific feature license, select **Cluster > Settings**.
2. Under **Licenses**, select →.
3. Select **Features**.
4. Locate the licensed feature you want to view and select ▼ to view the license details.

³ <https://docs.netapp.com/us-en/ontap/system-admin/install-license-task.html>

Delete a license

System Manager - ONTAP 9.8 and later System Manager - ONTAP 9.7 and earlier CLI

1. To delete a license, select **Cluster > Settings**.
2. Under **Licenses**, select [→](#).
3. Select **Features**.
4. Select the licensed feature you want to delete and **Delete legacy key**.

4

32. NetApp's Cluster Management ONTAP performs the step of storing license information for a computing cluster in a memory module associated with a computing node wherein the licensing information includes a bundle-type parameter that identifies a size of the cluster and at least one additional bundle-type characteristic of the cluster. For example, ONTAP stores license information for a computing cluster either at the factory or by adding them to the System Manager. By way of further example, the System Manager contains the license information names and serial numbers for all nodes in the cluster as well as the features of the licensed package.

License types

A package can have one or more of the following types of licenses installed in the cluster:

- Capacity license (ONTAP Select and FabricPool only)

An ONTAP Select instance is licensed according to the amount of data that the user wants to manage. For example, the user might buy a 10 TB capacity license to enable ONTAP Select to manage up to 10 TB of data. If more storage capacity is attached to the system than ONTAP Select is licensed to manage, ONTAP Select will not operate. By default, the maximum storage capacity that can be attached to an ONTAP Select instance is 2 TB until a capacity license (for example, a 5 TB capacity license, a 10 TB capacity license, and so on) is purchased and installed.

Starting with ONTAP 9.2, FabricPool-enabled aggregates require a capacity license to be used with a third-party storage tier (for example, AWS). The FabricPool capacity license defines the amount of data that can be stored in the cloud tier storage.

⁴ <https://docs.netapp.com/us-en/ontap/system-admin/manage-license-task.html>

- **Maximum Capacity**
 - For ONTAP Select, displays the maximum amount of storage that can be attached to the ONTAP Select instance.
 - For FabricPool, displays the maximum amount of third-party object store storage that can be used as cloud tier storage.

5

Package	Cluster/Node	Serial Number	Type
Cluster Base License	Cluster02	1-80-000008	Node Locked
CIFS License	Cluster02-02	1-81-00000000000000004034389062	Node Locked
FCP License	Cluster02-02	1-81-00000000000000004034389062	Node Locked
FlexClone License	Cluster02-02	1-81-00000000000000004034389062	Node Locked
CIFS License	Cluster02-01	1-81-00000000000000004082368507	Node Locked
FCP License	Cluster02-01	1-81-00000000000000004082368507	Node Locked
FlexClone License	Cluster02-01	1-81-00000000000000004082368507	Node Locked

33. NetApp's Cluster Management ONTAP performs the step of storing license information for a computing cluster in a memory module associated with a computing node wherein the licensing information includes a bundle-type parameter that identifies a size of the cluster and at least one additional bundle-type characteristic of the cluster, and one or more node license parameters that identifies a characteristic of a computing node. For example, the System Manager identifies the license package and features available for each computing node in the cluster.

⁵ https://github.com/NetAppDocs/ontap-system-manager-classic/blob/main/online-help-96-97/concept_licenses.adoc

Examples

The following example displays the information for license package NFS. NFS is unlicensed in the cluster and no action is necessary to reduce the entitlement risk. The nodes, cluster1-01 and cluster-02, are missing a node-locked license. The serial numbers for both nodes are also displayed.

```
cluster1::> system license entitlement-risk show
Package      Entitlement Risk Corrective Action
-----
NFS          unlicensed      None
Nodes Without a Node-Locked License
-----
cluster1-01      1-81-000000000000004073806282
cluster1-02      1-81-000000000000004073806283
```

The following example displays the information for license package CIFS. The cluster has high entitlement risk for CIFS. The command displays serial numbers for all nodes in the cluster. Both nodes are missing a node-locked CIFS license. Node with serial number 1-81-000000000000004073806282 has used CIFS feature in the past week, and the node with serial number 1-81-000000000000004073806283 has not used this feature in the past week.

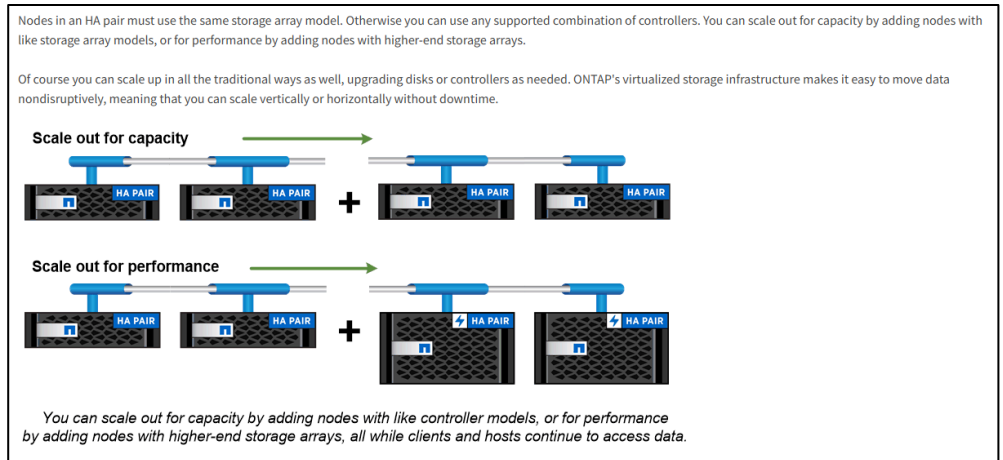
```
cluster1::> system license entitlement-risk show -detail
Package      Entitlement Risk Corrective Action
-----
CIFS          high          Acquire a node-locked license
Serial Numbers      Licensed Usage
-----
1-81-000000000000004073806282 false      true
1-81-000000000000004073806283 false      false
```

6

Package	Cluster/Node	Serial Number	Type
Cluster Base License	Cluster02	1-80-000008	Node Locked
CIFS License	Cluster02-02	1-81-000000000000004034389062	Node Locked
FCP License	Cluster02-02	1-81-000000000000004034389062	Node Locked
FlexClone License	Cluster02-02	1-81-000000000000004034389062	Node Locked
CIFS License	Cluster02-01	1-81-000000000000004082368507	Node Locked
FCP License	Cluster02-01	1-81-000000000000004082368507	Node Locked
FlexClone License	Cluster02-01	1-81-000000000000004082368507	Node Locked

34. NetApp's Cluster Management ONTAP performs the step of initializing the computing cluster in a first computing node.

⁶ <https://docs.netapp.com/us-en/ontap-cli/system-license-entitlement-risk-show.html#parameters>



7

Single-node clusters

A single-node cluster is a special implementation of a cluster running on a standalone node. You might want to deploy a single-node cluster in a branch office, for example, assuming the workloads are small enough and that storage availability is not a critical concern.

In this scenario, the single-node cluster would use SnapMirror replication to back up the site's data to your organization's primary data center. ONTAP Select, with its support for ONTAP running on commodity hardware, would be a good candidate for this type of implementation.

8

35. NetApp's Cluster Management ONTAP performs the step of adding one or more available computing nodes to the computer cluster. For example, once the cluster is provisioned initially, the cluster can be scaled in or out by adding or removing the nodes.

You can use ONTAP System Manager classic (available in ONTAP 9.7 and earlier) to increase the size and capabilities of your storage by adding compatible nodes to the cluster and configuring the node network details. You can also view the summary of the nodes.

When you log in to System Manager, System Manager automatically detects compatible nodes that have been cabled but have not been added to the cluster and prompts you to add the nodes. You can add compatible nodes as and when System Manager detects the nodes or you can manually add the nodes at a later time.

9

⁷ <https://docs.netapp.com/us-en/ontap/concepts/cluster-storage-concept.html#:~:text=Nodes%20in%20an%20HA%20pair,with%20higher-end%20storage%20arrays>

⁸ https://www.n0tes.fr/2022/11/03/NetApp-qtrees-volume-aggregate-luns/ONTAP_9_Documentation.pdf

⁹ https://github.com/NetAppDocs/ontap-system-manager-classic/blob/main/online-help-96-97/concept_cluster_expansion.adoc

36. NetApp's Cluster Management ONTAP performs the step of activating the computing cluster when the computing cluster includes a number of nodes that comply with the bundle-type parameter.

• Standard license (`license`)

A standard license is a node-locked license. It is issued for a node with a specific system serial number (also known as a *controller serial number*). A standard license is valid only for the node that has the matching serial number.

Installing a standard, node-locked license entitles a node to the licensed functionality. For the cluster to use licensed functionality, at least one node must be licensed for the functionality. It might be out of compliance to use licensed functionality on a node that does not have an entitlement for the functionality.

• Site license (`site`)

A site license is not tied to a specific system serial number. When you install a site license, all nodes in the cluster are entitled to the licensed functionality. The `system license show` command displays site licenses under the cluster serial number.

If your cluster has a site license and you remove a node from the cluster, the node does not carry the site license with it, and it is no longer entitled to the licensed functionality. If you add a node to a cluster that has a site license, the node is automatically entitled to the functionality granted by the site license.

10

37. Defendant directly infringes the '416 Patent through its implementation and use of NetApp's Cluster Management ONTAP, performing at least the steps of claim 1.

38. Defendant has indirectly infringed and continues to indirectly infringe one or more claims of the '416 Patent by knowingly and intentionally inducing others, including NetApp customers and end-users, to directly infringe, either literally or under the doctrine of equivalents, by making, using, offering to sell, selling, and/or importing into the United States products that include the infringing technology.

39. Defendant, with knowledge that these products, or the use thereof, infringe the '416 Patent at least as of December 15, 2015,¹¹ knowingly and intentionally induced, and continues to knowingly and intentionally induce, direct infringement of the '416 Patent by providing these

¹⁰ <https://docs.netapp.com/us-en/ontap/system-admin/license-types-licensed-method-concept.html>

¹¹ Plaintiffs sent Defendant correspondence on April 7, 2021, notifying them of their infringement of the '416 Patent. Thereafter, Plaintiffs and Defendant exchanged emails beginning in May 2021 regarding Defendant's infringement of the '416 Patent and the other Asserted Patents. Therefore, Defendant has had knowledge of the '416 Patent as of at least April 2021. Defendant also cited to the '416 Patent family in relation to its own U.S. Patent Application No. 13/359,203 filed on January 26, 2012, which issued as U.S. Patent No. 9,203,900 on December 15, 2015. Therefore, on information and belief, Defendant had knowledge of the '416 Patent family at least as early as December 15, 2015.

products to end-users for use in an infringing manner. Additionally, on information and belief, Defendant has adopted a policy of not reviewing the patents of others, and instructing its officers, agents, and employees to not review the patents of others, including specifically those related to Defendant's specific industry, with the subjective belief that there was a high probability that Defendant would learn of its infringing activities, thereby remaining willfully blind to the '416 Patent at least as early as the issuance of the '416 Patent

40. Defendant has induced infringement by others, including end-users, with the intent to cause infringing acts by others or, in the alternative, with the belief that there was a high probability that others, including end-users, infringe the '416 Patent, but while remaining willfully blind to the infringement. Defendant has induced, and continues to induce, infringement by their customers and end-users by supplying them with instructions on how to operate the infringing technology in an infringing manner, while also making publicly available information on the infringing technology via Defendant's website, product literature and packaging, and other publications.¹²

41. Plaintiffs have suffered damages as a result of Defendant's direct and indirect infringement of the '416 Patent in an amount to be proven at trial.

42. Plaintiffs have suffered, and will continue to suffer, irreparable harm as a result of Defendant's infringement of the '416 Patent, for which there is no adequate remedy at law, unless Defendant's infringement is enjoined by this Court.

43. Defendant has committed and continues to commit acts of infringement that Defendant actually knew or should have known constituted an unjustifiably high risk of

¹² See, e.g., <https://docs.netapp.com/us-en/ontap/index.html>; see also the supporting proof referenced herein

infringement of at least one valid and enforceable claim of the '416 Patent. Defendant's direct and indirect infringement of the '416 Patent has been and continues to be willful, intentional, deliberate, and/or in conscious disregard of rights under the patent. Plaintiffs are entitled to an award of treble damages, reasonable attorney fees, and costs in bringing this action.

COUNT II
(Infringement of the '571 Patent)

44. Paragraphs 1 through 25 are incorporated by reference as if fully set forth herein.

45. Plaintiffs have not licensed or otherwise authorized Defendant to make, use, offer for sale, sell, or import any products that embody the inventions of the '571 Patent.

46. Defendant has infringed, and continues to directly infringe, the '571 Patent, either literally or under the doctrine of equivalents, without authority and in violation of 35 U.S.C. § 271, by making, using, offering to sell, selling, and/or importing into the United States products that satisfy each and every limitation of one or more claims of the '571 Patent. Such products include systems using NetApp SANtricity on E-Series controllers.

47. For example, Defendant has infringed, and continues to directly infringe, at least claim 12 of the '571 Patent by making, using, offering to sell, selling, and/or importing into the United States products that include NetApp SANtricity on E-Series controllers.

48. NetApp SANtricity on E-Series controllers performs a method for use with a cluster of storage controllers that are able to access at least one storage subsystem having a storage volume. For example, NetApp SANtricity on E-Series controllers provide methods for managing and accessing storage volumes.

1 Introduction

NetApp® E-Series storage arrays are often deployed in cluster environments that require high-bandwidth and low-latency performance. However, volume ownership thrashing and other multipath/failover issues have been longstanding challenges in clustered environments, especially when hosts running different OSs access the same storage volumes on the array.

Over the past few years, NetApp has released enhancements in the E-Series firmware that improve interoperability between hosts and E-Series arrays in heterogeneous host OS clustered environments. These changes have culminated in a new failover mode that greatly improves cluster interoperability, and these most recent enhancements are the next step in an ongoing evolution. This includes two new enhancements released in April 2017 that are designed to specifically resist cluster volume ownership thrash.

Options have existed in the past that allow multipath failover to be avoided in deployments where that option made sense, but now these new options exist to build cluster solutions without risk of ownership thrash while still retaining failover and host I/O access redundancy. For some limited configurations, options also exist to provide limited failback in clustered environments. This enables automatic recovery software to move LUN ownership back to preferred paths following the correction of SAN connectivity faults, again without the risk of LUN ownership thrash.

13

The solution

Create a storage infrastructure that is fast, simple, and reliable. The NetApp® SANtricity® OS flexibly delivers performance for a wide variety of workloads while keeping data secure and available. NetApp E-Series systems and SANtricity software make it easy, thanks to industry-leading performance, reliability, and simplicity of administration.

SANtricity OS

With the SANtricity OS, you can perform all your management tasks while the storage remains online, with complete read and write data access. This capability means that storage administrators can make configuration changes, perform maintenance, and expand storage capacity without scheduling downtime.

14

Beginning with NetApp SANtricity® OS 11.30 and controller firmware version 08.30.XX.XX, NetApp E-Series storage arrays offer automatic load balancing (ALB), a feature that monitors incoming I/O workload and dynamically makes configuration changes to balance the workload across the array's two I/O controllers. This document is an overview of the behavior of the ALB feature, its key configuration parameters, and its host interoperability enhancements.

15

¹³ <https://www.netapp.com/media/17097-tr4604.pdf>

¹⁴ <https://www.netapp.com/pdf.html?item=/media/7676-ds-3891.pdf>

¹⁵ <https://www.netapp.com/media/17144-tr4737.pdf>

Hybrid Flash (E-Series)					
Model	E5760	E5724	E2860	E2824	E2812
Front View					
Max Capacity Decimal Binary	4.8PB 4.23PiB	4.57PB 4.06PiB	1.8PB 1.6PiB	1.76PB 1.56PiB	1.8PB 1.6PiB
Max Dynamic Disk Pool Volumes	2048	2048	512	512	512
Max Volumes	2048	2048	512	512	512
Max Vol Size for Dynamic Disk Pool	64TB	64TB	64TB	64TB	64TB
Max Disk Pools	20	20	20	20	20
Max Drive Quantity	480	444	180	180	180
Environmental all values based on factory configurations	Rack Units	4U	2U	2U	2U
	Weight English Metric	217.5lb 98.6kg	50.6lb 22.9kg	224.7lb 101.9kg	50.6lb 22.9kg
	Amps ³ Typical Worst-case	6.10 7.68 @220V	4.66 6.40 @110V 2.09 2.96 @220V	5.78 7.49 @220V	4.18 5.92 @110V 2.09 2.96 @220V
	BTU/hr ³ Typical Worst-case	4529 5705 @110V, 220V	1836 2483 @110V, 220V	4593 5641 @110V, 220V	1553 2200 @110V, 220V
Platform Specifications all values based on factory configurations	Processor	2x 64-bit 16 total cores	2x 64-bit 16 total cores	2x 64-bit 8 total cores	2x 64-bit 8 total cores
	Memory	64GB 16GB	64GB 16GB	64GB 16GB	64GB 16GB
	Host Interface Cards	2	2	2	2
	Onboard FC/iSCSI Ports	4x SFP+ 1/10GbE or 16Gb FC	4x SFP+ 1/10GbE or 16Gb FC	4x 10GbE/16Gb SFP+ or 4x 10GbE RJ45	4x 10GbE/16Gb SFP+ or 4x 10GbE RJ45
	Onboard SAS Ports	4x 12Gb miniSAS HD	4x 12Gb miniSAS HD	4x 12Gb miniSAS HD	4x 12Gb miniSAS HD
Min SANtricity Firmware Version		11.40	11.40	11.30.1	11.30

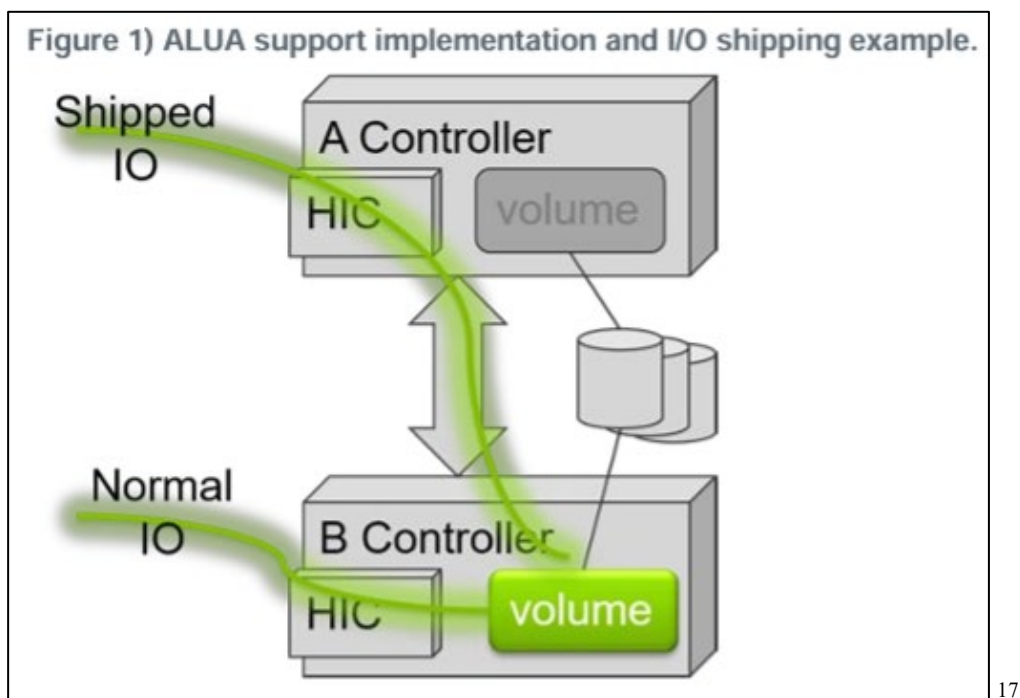
16

49. NetApp SANtricity on E-Series controllers performs the step of receiving, by a first of the storage controllers that is currently an owner of the storage volume, client requests from one or more client computers and proxy requests from one or more other storage controllers in the cluster. For example, controllers of the NetApp E-Series receive requests from clients (e.g., normal IO requests) as well as from other controllers (e.g., shipped IO requests).

¹⁶ <https://hww.netapp.com/Resources/Posters/HWU-2017-NA00-0494-000-000.pdf>

To achieve consistent, ultralow latency, the E-Series firmware implements a highly optimized write data cache coherency model and a lean I/O path that uses the notion of volume ownership. This means that I/O for a given volume must be processed by the controller that owns that specific volume. This design greatly simplifies the data-caching implementation and hardware design, minimizing I/O latency while keeping the overall product cost low.

Despite using a design that employs an ownership model, E-Series arrays also have multiple target ports on each I/O controller, either on a host interface card (HIC) or on the controller baseboard itself. Both HIC and baseboard ports support an active-active access model. With this model, hosts can access all volumes through any target port on either controller, regardless of volume ownership. This access is accomplished with an "I/O shipping" mechanism, in which I/O received on the nonowning controller is processed by the owning controller using intercontroller messaging. For example, Figure 1 shows the case where a volume is owned by the B controller. I/O directed by the host to the A controller is first shipped through the controller interconnect channels to the B controller, then processed.



17

50. NetApp SANtricity on E-Series controllers performs the step of, based on numbers of the client requests and the proxy requests, determining that a transfer of ownership is to be performed from the first storage controller to a second storage controller in the cluster. For example, the E-Series automatic load balancing and redundant Host/Access/Failover features determines the transfer of volume ownership between controllers based on the workload I/O dynamically.

¹⁷ <https://www.netapp.com/media/17097-tr4604.pdf>

The Automatic Load Balancing feature provides automated I/O balancing and ensures that incoming I/O traffic from the hosts is dynamically managed and balanced across both controllers.

The Automatic Load Balancing feature provides improved I/O resource management by reacting dynamically to load changes over time and automatically adjusting volume controller ownership to correct any load imbalance issues when workloads shift across the controllers.

The workload of each controller is continually monitored and, with cooperation from the multipath drivers installed on the hosts, can be automatically brought into balance whenever necessary. When workload is automatically re-balanced across the controllers, the storage administrator is relieved of the burden of manually adjusting volume controller ownership to accommodate load changes on the storage array.

When Automatic Load Balancing is enabled, it performs the following functions:

- Automatically monitors and balances controller resource utilization.
- Automatically adjusts volume controller ownership when needed, thereby optimizing I/O bandwidth between the hosts and the storage array.

18

E-Series storage administration tools allow the storage administrator to assign ownership of each volume to one of the two array I/O controllers. This flexibility can be combined with the host multipath driver behavior of routing I/O for a given volume to the owning controller, which gives the storage administrator a method to control host I/O routing to the array. If the storage administrator also understands expected I/O workload to each volume, they can achieve a rudimentary workload balance by carefully distributing volume ownership across the two I/O controllers in the array.

The primary challenge facing the storage administrator in accomplishing this kind of manual workload balance across the two array controllers is that workload on each individual volume is often very dynamic. Therefore, an ownership distribution that accomplishes a satisfactory workload balance at a given point in time often doesn't continue to provide that balance for very long. In addition, upon initial creation of a new storage volume on the array, the storage administrator often doesn't know what the host I/O workload will be on that volume. Therefore, selecting initial controller ownership for a new volume is often difficult.

The ALB feature addresses these shortfalls through the automated volume ownership distribution process described later in this document.

Feature benefits

The primary goal of ALB is to remove the need for the storage administrator to manually manage volume ownership and create balanced workloads across the array I/O controllers. This automation allows simple storage management of E-Series arrays. It also provides best-in-class price/performance, made possible by the use of a lean I/O path and simple data cache coherency algorithms associated with the volume ownership model.

19

- **Application-specific** — When you are creating volumes using an application-specific workload, the system may recommend an optimized volume configuration to minimize contention between application workload I/O and other traffic from your application instance. Volume characteristics like I/O type, segment size, controller ownership, and read and write cache are automatically recommended and optimized for workloads that are created for the following application types.

20

51. NetApp SANtricity on E-Series controllers performs the step of, based on numbers of the client requests and the proxy requests, determining that a transfer of ownership is to be

¹⁸ <https://docs.netapp.com/us-en/e-series-santricity/sm-storage/what-is-automatic-load-balancing.html>

¹⁹ <https://www.netapp.com/media/17144-tr4737.pdf>.

²⁰ <https://docs.netapp.com/us-en/e-series/vcenter-plugin/vc-olh-how-does-my-selected-workload-impact-volume-creation.html>

performed from the first storage controller to a second storage controller in the cluster, wherein the determining comprises determining that the second storage controller has provided a higher number of proxy requests for the storage volume than the number of the client requests. For example, E-Series load balancing and failover features distribute ownership based on the number I/O requests from clients and controller. An example below shows an event generated when more than 75% of the I/O to a given volume is sent to the non-owning controller (proxy requests).

Automatic volume ownership distribution

ALB automates the process of balancing workload across the two I/O controllers in the array by periodically evaluating I/O workload to each volume and automatically selecting a volume ownership distribution that maximizes workload balance. The periodic rebalancing process is as follows:

1. I/O workload to each volume is monitored over a period of time, and I/O workload to each volume is predicted over the next time period using the collected workload data.
2. An optimal distribution of volume ownership across the two I/O controllers is determined to balance total workload on each controller in the next time period.
3. Transfer of ownership on a minimum number of volumes is automatically initiated to achieve a workload balance in the upcoming time period.
4. Those volume ownership changes are communicated to the host multipath drivers to cause the hosts to direct I/O to the array controllers according to the new ownership distribution.

ALB periodically initiates this reevaluation and rebalancing process. It can therefore react to the dynamic I/O workload distribution across volumes that is typical in enterprise data centers and automatically adjust to maintain the maximum performance of the array.

I/O workload statistics gathered by ALB consider several aspects of workload:

- The number of I/O requests being received from all hosts per unit of time (IOPS).
- The total size of I/O requests being received from all hosts per unit of time (bandwidth).
- Utilization of controller write data cache memory. This considers the read/write mix in the incoming workload to make sure that both controllers have a similar level of total-write-cache utilization.
- Total workload (IOPS and bandwidth) sent to the drives per unit of time as a result of the incoming host I/O. This is a normalized workload factor that considers the RAID level and stripe width of the volumes receiving I/O from the hosts. It also considers differences in the cost of read and write operations, the number of full RAID-stripe writes, the cost of write data cache mirroring, and so on.

Appendix A - Major event log content

The following major event log (MEL) events are generated by ALB, target failback, and SAN connectivity reporting.

Volume transfer events

These informational events pertain to volumes that have transferred ownership as part of a failover, failback, or ALB workload rebalance.

MEL_EV_LOAD_BALANCING_VD_TRANSFER

Event ID: 0x2044 **Priority:** INFO

Event Description: IO shipping implicit volume transfer

This event is logged when the array controller transfers ownership of a volume because it detects that more than 75% of the I/O to a given volume is being sent to the non-owning controller. This is most typically the failover case for which disruptions in the SAN force one or more hosts to lose all connectivity to the owning controller. This can also happen, however, when ownership is changed, but the host multipath driver does not observe the ownership change notification and then fails to redirect I/O to the new owning controller. See section 0, "Implicit ownership changes – Host follow-over failures" for details.

Workload balance objectives

ALB considers several different (sometimes competing) objectives when determining the optimal volume ownership distribution across the two I/O controllers. These objectives include:

- Achieving a balance of the measured workload factors (noted earlier in this section) across the two I/O controllers.
- Minimizing (and eliminating if possible) any split of ownership between the two controllers for volumes of RAID level 0, 1, 3, 5, or 10 that are using the same HDDs for storage (for example, volumes in the same drive group). Note that volumes that are part of a dynamic disk pool or that are backed by SSD storage are not considered in this objective. The intent of this objective is to minimize head seek time on HDDs when flushing write data cache to media. Excess head seek time can occur when both controllers contain write data in cache destined for the same set of drives.
- Minimizing the total cost of realigning volume ownership for load balancing purposes. This includes the following issues:
 - Selecting a new distribution of volume ownership that achieves a satisfactory workload balance, but that at the same time minimizes the number of volumes that must change ownership.
 - Avoiding transferring ownership of any volumes that have recently transferred ownership, either for workload distribution purposes, for failover or failback, or for any other reason. This minimizes volume ownership thrash caused by load balancing.

2.4 Explicit Versus Implicit ALUA/TPGS and Failover

T10-SPC defines two methods of ALUA support: implicit and explicit. Implicit ALUA means the array controller manages the asymmetric access states of the target port groups (that is, the array manages the volume ownership, which in turn determines the asymmetric access state of each target port group). This is accomplished by monitoring all incoming read/write requests to a given volume on the array and determining which controller is receiving the bulk of the incoming requests. In the absence of any SAN connectivity fault, the preferred owner and current owner of a given volume are the same controller. It tends to stay that way because the host multipath driver directs I/O at the current owning controller (active/optimized target port group) in order to maximize performance. When a fault occurs such that hosts with access to the volume lose connection to the preferred/current controller, I/O is directed at the nonowning/nonpreferred controller, and the array decides to move current ownership to the nonpreferred

controller to minimize the I/O shipping penalty. By definition, this means more than 75% of the I/O to a given volume over a 5-minute period arrived at the nonowning controller. The result is an implicit failover, and the current and preferred owner are now different.

21

52. NetApp SANtricity on E-Series controllers performs the step of performing the transfer of ownership of the storage volume from the first storage controller to the second storage controller in response to the determining. For example, upon determining a need for changing a volume ownership, NetApp controllers execute the change. The ability to automatically change volume ownership is enabled by default in newer arrays running SANtricity and E-Series controller firmware.

Enabling and disabling ALB

ALB can be enabled or disabled with a user configuration setting available in both SANtricity System Manager and SMcli. Disabling the feature stops the periodic collection of workload data and the evaluation of workload balance across controllers. It also eliminates any volume ownership transfers that were initiated solely for the purpose of balancing the workload. Prior to SANtricity OS 11.40.1 and controller firmware 08.42.XX.XX, disabling ALB also disables enhanced SAN connectivity reporting.

New arrays running SANtricity OS 11.30 and E-Series controller firmware 08.30.XX.XX or later from the factory have ALB enabled by default upon initial startup. When the array controller firmware is upgraded to 08.30.XX.XX or later for the first time, it might determine that there is no existing setting for the ALB feature. It then checks to see whether any user-defined volumes are configured on the array. If not, ALB is enabled by default.

ALB attempts to redistribute volume ownership once per hour to optimize workload balance across the two I/O controllers. It gathers I/O workload statistics in 5-minute intervals throughout the hour and uses that near-term historical data to predict a forward-looking balance of workload based on volume ownership distribution at the time of evaluation. If that workload balance is outside of programmatically determined thresholds, ALB then identifies a new optimized distribution of volume ownership that

achieves a balance between the two I/O controllers within thresholds. The selection process attempts to minimize the total cost of changing volume ownership by minimizing the total number of volumes that need to move.

²¹ <https://www.netapp.com/media/17144-tr4737.pdf>

Automatic volume ownership distribution

ALB automates the process of balancing workload across the two I/O controllers in the array by periodically evaluating I/O workload to each volume and automatically selecting a volume ownership distribution that maximizes workload balance. The periodic rebalancing process is as follows:

1. I/O workload to each volume is monitored over a period of time, and I/O workload to each volume is predicted over the next time period using the collected workload data.
2. An optimal distribution of volume ownership across the two I/O controllers is determined to balance total workload on each controller in the next time period.
3. Transfer of ownership on a minimum number of volumes is automatically initiated to achieve a workload balance in the upcoming time period.
4. Those volume ownership changes are communicated to the host multipath drivers to cause the hosts to direct I/O to the array controllers according to the new ownership distribution.

ALB periodically initiates this reevaluation and rebalancing process. It can therefore react to the dynamic I/O workload distribution across volumes that is typical in enterprise data centers and automatically adjust to maintain the maximum performance of the array.

22

53. Defendant directly infringes the '571 Patent through its implementation and use of NetApp SANtricity on E-Series controllers, performing at least the steps of claim 12.

54. Defendant has indirectly infringed and continues to indirectly infringe one or more claims of the '571 Patent by knowingly and intentionally inducing others, including NetApp customers and end-users, to directly infringe, either literally or under the doctrine of equivalents, by making, using, offering to sell, selling, and/or importing into the United States products that include the infringing technology.

55. Defendant, with knowledge that these products, or the use thereof, infringe the '571 Patent at least as of November 10, 2015,²³ knowingly and intentionally induced, and continues to knowingly and intentionally induce, direct infringement of the '571 Patent by providing these products to end-users for use in an infringing manner. Additionally, on information and belief,

²² <https://www.netapp.com/media/17144-tr4737.pdf>

²³ Plaintiffs sent Defendant correspondence on June 6, 2022, notifying them of their infringement of the '571 Patent. Thereafter, Plaintiffs and Defendant exchanged emails regarding Defendant's infringement of various patents owned by Plaintiff. Defendant cited directly to the '571 Patent in relation to its own U.S. Patent Application No. 14/480,273, which was filed on August 8, 2014, and issued as U.S. Patent No. 9,182,918 on November 10, 2015. Therefore, on information and belief, Defendant had knowledge of the '571 Patent at least as of November 10, 2015.

Defendant has adopted a policy of not reviewing the patents of others, and instructing its officers, agents, and employees to not review the patents of others, including specifically those related to Defendant's specific industry, with the subjective belief that there was a high probability that Defendant would learn of its infringing activities, thereby remaining willfully blind to the '571 Patent at least as early as the issuance of the '571 Patent

56. Defendant has induced infringement by others, including end-users, with the intent to cause infringing acts by others or, in the alternative, with the belief that there was a high probability that others, including end-users, infringe the '571 Patent, but while remaining willfully blind to the infringement. Defendant has and continues to induce infringement by their customers and end-users by supplying them with instructions on how to operate the infringing technology in an infringing manner, while also making publicly available information on the infringing technology via Defendant's website, product literature and packaging, and other publications.²⁴

57. Plaintiffs have suffered damages as a result of Defendant's direct and indirect infringement of the '571 Patent in an amount to be proven at trial.

58. Plaintiffs have suffered, and will continue to suffer, irreparable harm as a result of Defendant's infringement of the '571 Patent, for which there is no adequate remedy at law, unless Defendant's infringement is enjoined by this Court.

59. Defendant has committed and continues to commit acts of infringement that Defendant actually knew or should have known constituted an unjustifiably high risk of infringement of at least one valid and enforceable claim of the '571 Patent. Defendant's direct and indirect infringement of the '571 Patent has been and continues to be willful, intentional,

²⁴ See, e.g., <https://www.netapp.com/data-management/santricity/>; see also the supporting proof referenced herein.

deliberate, and/or in conscious disregard of rights under the patent. Plaintiffs are entitled to an award of treble damages, reasonable attorney fees, and costs in bringing this action.

COUNT III
(Infringement of the '686 Patent)

60. Paragraphs 1 through 25 are incorporated by reference as if fully set forth herein.

61. Plaintiffs have not licensed or otherwise authorized Defendant to make, use, offer for sale, sell, or import any products that embody the inventions of the '686 Patent.

62. Defendant has infringed, and continues to directly infringe, the '686 Patent, either literally or under the doctrine of equivalents, without authority and in violation of 35 U.S.C. § 271, by making, using, offering to sell, selling, and/or importing into the United States products that satisfy each and every limitation of one or more claims of the '686 Patent. Such products include HDFS.

63. For example, Defendant has infringed, and continues to directly infringe, at least claim 1 of the '686 Patent by making, using, offering to sell, selling, and/or importing into the United States products that include HDFS.

64. On information and belief, NetApp uses HDFS in the U.S.

Naren Yadav

Solution Architect/Service Support Manager - NetApp

San Ramon, CA

-Email me on Indeed: <http://www.indeed.com/r/Naren-Yadav/edaba424071a9aea>

Authorized to work in the US for any employer

Work Experience

Service Support Manager

NetApp

2011 to Present

Lead business reviews with IT and business user executives; vendor management with focus on sustainable Service

Level Agreements.

- Motivate and advise Business Process team in business reengineering projects.
- Projects to reduce downtime and lower storage and operating expenses via NetApp Open Solution for Hadoop enterprise-class infrastructures.

• Multiple technologies including Cloudera CDH 4.1 with flume/Hbase and DataStax Solr; also hdfs, Hbase, Hive,

Cassandra DataStax Solr and REST APIs.

- ETL extracts from Hadoop to relational databases using Pentaho ETL.
- Solution design and 24x7 support for Big Data solutions for Tier 1 and Tier 2 applications.

25

Simon Kwan

Sr. Software Engineer - NetApp Inc

Fremont, CA

-Email me on Indeed: <http://www.indeed.com/r/Simon-Kwan/c5511ddd9a2d1952>

Professional Summary

A collaborative software engineering professional with substantial experience designing and executing solutions for complex business problems that involve large-scale distributed systems and data. Demonstrated ability to adapt to fast changing environment and deliver results on schedule under pressure. Research and evaluate emerging technologies for projects.

Authorized to work in the US for any employer

Work Experience

Sr. Software Engineer

NetApp Inc

2008 to Present

- Production Batch(PB) System - A business critical MapReduce(MR) jobs system running on multiple 100+ nodes clusters to process NetApp worldwide customers' AutoSupport(ASUP) data that contains configuration details, vital status, and performance information. Roughly 200k systems phone home in total of 500 GB to 1 TB ASUP data daily. Receive employee award for this project.

* lead Java development utilizing Rules Engine in MR to analyze, calculate, and predict system healthiness with rules from At Risk Systems, Storage Efficiency and other business domains; develop Java data structures and access logic

* Translate Subject Matter Experts(SME) business requirements into rules then implement them in Hadoop Streaming with Python, and Rules Engine domain specific language DRL. Enhance objects Avro schema; code logic to ETL data from HDFS and HBase for model objects

26

²⁵ <http://www.indeed.com/r/Naren-Yadav/edaba424071a9aea>

²⁶ <http://www.indeed.com/r/Simon-Kwan/c5511ddd9a2d1952>

Abhishek Sai

Hadoop Administrator - NetApp Inc

Sunnyvale, CA

-Email me on Indeed: <http://www.indeed.com/r/Abhishek-Sai/343b24045b1f8b52>

Authorized to work in the US for any employer

Work Experience

Hadoop Administrator

NetApp-Sunnyvale, CA

November 2016 to Present

- Five years of experience on Big Data Technologies and database systems.
- 4+ years of experience as Hadoop Administrator and 4 years of Red Hat Enterprise Linux Administration.
- Experience in installation, configuration, upgrading and managing CDH Hadoop clusters from the scratch.
- Experience in upgrading CDH cluster from CDH5 to CDH6.
- Implemented in setting up standards and processes for Hadoop based application design and implementation.
- Good knowledge of installing and managing distributions of Hadoop like CDH6, CDH5, Hortonworks and vanilla Hadoop.
- Well versed supporting users on the CDH Big data platform in different environments.
- Implementation of High Availability for HDFS, Resource manager, Hue.

27

Software and Automation Developer

Jan 2020 - May 2023

NetApp Inc - Morrisville, NC

- Created a time-series based Splunk UI analytics dashboard providing daily estimates of cloud savings (averaging \$50k monthly) on enabling cloud governance on R&D activities across all 3 public cloud providers, a big push for company executives to leverage house cloud governance tools.
- Reduced pipeline deployment failures by 90% using a rearchitected pipeline.
- Brought down the average CI/CD deployment times to 30 minutes combining dev, stage and production which averaged 8 hours with the legacy pipeline.
- Implemented Kubernetes Secrets to securely store and manage sensitive data such as API keys, credentials, and certificates.
- Developed Python based automation to perform governance activities on AWS, AZURE and GCP including but not limited to shutdown/termination of testbeds or idle/unused computing instances/virtual machines.
- Developed and deployed containerized Python based microservices performing VMWare & Openstack VM operations to Kubernetes.
- Developed and optimized ETL pipelines using PySpark and Pandas, improving data processing speed.
- Implemented SparkSQL queries to transform large datasets for reporting and analytics, reducing query execution time.
- Designed, developed, and maintained Airflow managed data pipelines that ingested billing data extracted from cloud providers using REST, then transformed and loaded them into MySQL, HDFS file systems(parquet) to build Tableau reports.
- Designed and developed a serverless framework in AWS using Lambdas and Step Functions to create a deployment workflow to provision/unprovision VM's in the downstream on-prem OpenStack environment.
- Implemented IaC using well-structured Ansible scripts for deploying to both AWS and on-prem Kubernetes clusters.

28

²⁷ <http://www.indeed.com/r/Abhishek-Sai/343b24045b1f8b52>

²⁸ <https://www.linkedin.com/in/pratyush-mulpuru-97453915/>

Hadoop Developer

NetApp-Sunnyvale, CA

February 2015 to Present

Project Name: Customer Knowledge Platform (CKP)

NetApp is a leading integrated retailer with almost 2,500 full-line and specialty retail stores in the United States and Canada. NetApp is the leading home appliance retailer as well as a leader in tools, lawn and garden, fitness equipment and automotive repair and maintenance. I was a part of Big Data Processing Team to take advantage of available user data to make better decisions that significantly enhanced organizational success. I was involved in setting up Cloudera Hadoop Cluster and wrote MapReduce jobs and Hive queries to explore through the data of customer sales to find significant information for trend analysis.

Responsibilities:

- Involved in start to end process of Hadoop cluster installation, configuration and monitoring.
- Responsible for building scalable distributed data solutions using Hadoop
- Installed and configured Hive, Pig, Sqoop and Oozie on the Hadoop cluster.
- Developed Pig Latin scripts to extract the data from the web server output files to load into HDFS.
- Handled importing of data from various data sources, performed transformations using Hive, MapReduce, loaded data into HDFS and Extracted the data from Oracle into HDFS using Sqoop.
- Optimized Map/Reduce Jobs to use HDFS efficiently by using various compression mechanisms.
- Analyzed the data by performing Hive queries and running Pig scripts to study customer behavior.
- Installed Oozie workflow engine to run multiple Hive and Pig jobs.
- Developed workflow in Oozie to automate the tasks of loading the data into HDFS and pre-processing with Pig.
- Responsible for managing and reviewing Hadoop log files.
- Designed and developed data management system using MySQL.
- Developed entire frontend and backend modules using Python on Django Web Framework.
- Wrote Python scripts to parse XML documents and load the data in database..
- Developed dynamic webpages using HTML, CSS, and JavaScript.
- Worked on CSV files while trying to get input from the MySQL database.
- Exported the analyzed data to the relational databases using Sqoop for visualization and to generate reports for the BI team.
- Wrote scripts to automate application deployments and configurations. Hadoop cluster performance tuning and monitoring.
- Worked on NoSQL databases including HBase.
- Worked on importing and exporting data from Oracle and DB2 into HDFS and HIVE using Sqoop.

Environment: HDFS, Hive, PIG, UNIX, SQL, Java MapReduce, Hadoop Cluster, HBase, Sqoop, Oozie, Linux, Cloudera Hadoop Distribution, Python, MySQL.

29

65. The HDFS performs a method of providing data security for use with a file system.

For example, the Apache Hadoop framework provides security features to protect data present in Hadoop clusters. Hadoop uses HDFS (a file system) to store data. HDFS comprises a name node and multiple data nodes.

Hadoop, an open-source framework used for distributed storage and processing of big data, provides various security features to protect data and ensure secure access to Hadoop clusters. Here are some of the ways Hadoop is secured:

²⁹ <http://www.indeed.com/r/Sr-Hadoop/5eca63ccd32a48b0>

2. Authorization: Hadoop uses Access Control Lists (ACLs) to manage permissions for users and groups accessing data on the Hadoop Distributed File System (HDFS). Additionally, Hadoop provides fine-grained authorization through the use of Apache Ranger, which allows administrators to create policies that define access control for specific resources and actions.

30

NameNode and DataNodes

HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.

31

The Communication Protocols

All HDFS communication protocols are layered on top of the TCP/IP protocol. A client establishes a connection to a configurable TCP port on the NameNode machine. It talks the ClientProtocol with the NameNode. The DataNodes talk to the NameNode using the DataNode Protocol. A Remote Procedure Call (RPC) abstraction wraps both the Client Protocol and the DataNode Protocol. By design, the NameNode never initiates any RPCs. Instead, it only responds to RPC requests issued by DataNodes or clients.

32

66. The HDFS performs the step of applying a mapping function to data block numbers that are associated with a file, wherein the data block numbers are contained in an index node associated with said file. For example, NameNode performs various operations, such as opening and closing of files, and mapping data block locations (e.g., data block number). Mapping (e.g., mapping function) is the process of determining which data block is stored at which location.

³⁰ https://medium.com/@big_data_landscape/hadoop-security-5dac0ffe209e

³¹ http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html#NameNode_and_DataNodes

³² <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>

If you take another look at Figure 4-3, you can see the NameNode daemon running on a master node server. All mapping information dealing with the data blocks and their corresponding files is stored in a file named `fsimage`. HDFS is a journaling file system, which means that any data changes are logged in an edit journal that tracks events since the last *checkpoint* — the last time when the edit log was merged with `fsimage`. In HDFS, the edit journal is maintained in a file named `edits` that's stored on the NameNode.

The client opens the file it wishes to read by calling `open()` on the `FileSystem` object, which for HDFS is an instance of `DistributedFileSystem` (step 1 in Figure 3-2). `DistributedFileSystem` calls the namenode, using remote procedure calls (RPCs), to determine the locations of the first few blocks in the file (step 2). For each block, the namenode returns the addresses of the datanodes that have a copy of that block. Furthermore, the datanodes are sorted according to their proximity to the client (according to the topology of the cluster's network; see “Network Topology and Hadoop” on page 70). If the client is itself a datanode (in the case of a MapReduce task, for instance), the client will read from the local datanode if that datanode hosts a copy of the block (see also Figure 2-2 and “Short-circuit local reads” on page 308).

33

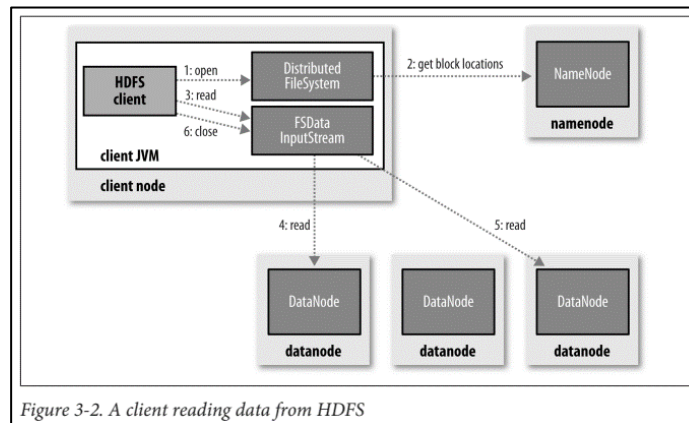


Figure 3-2. A client reading data from HDFS

34

³³<https://ia800201.us.archive.org/2/items/HadoopForDummiesDirkDeRoos2014/Hadoop%20For%20Dummies%20-%20%20Dirk%20deRoos%202014.pdf>, at Page 74 of 411.

³⁴ <https://grut-computing.com/HadoopBook.pdf>, at Page 97 of 756.

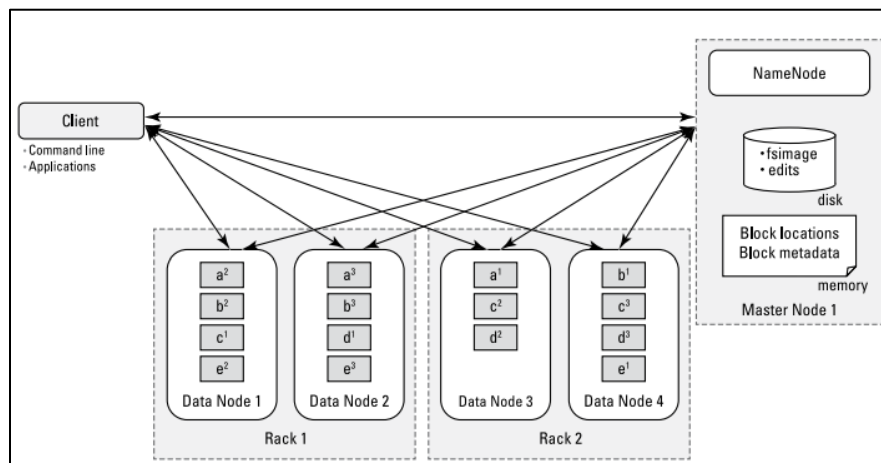
NameNode and DataNodes

HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.

35

By way of further example, data blocks are associated with a file. The data block locations are stored in the NameNode's (e.g., index node) memory.

After the startup process is completed, the NameNode has a complete picture of all the data stored in HDFS, and it's ready to receive application requests from Hadoop clients. As data files are added and removed based on client requests, the changes are written to the slave node's disk volumes, journal updates are made to the `edits` file, and the changes are reflected in the block locations and metadata stored in the NameNode's memory (see Figure 4-4).



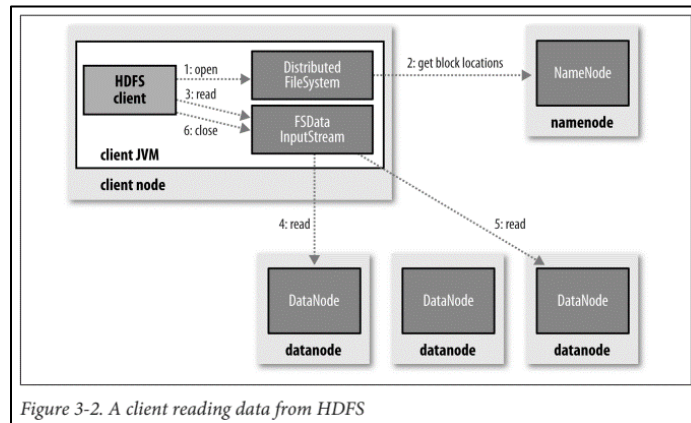
³⁵ <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>

Throughout the life of the cluster, the DataNode daemons send the NameNode heartbeats (a quick signal) every three seconds, indicating they're active. (This default value is configurable.) Every six hours (again, a configurable default), the DataNodes send the NameNode a block report outlining which file blocks are on their nodes. This way, the NameNode always has a current view of the available resources in the cluster.

36

67. The HDFS performs the step of obtaining mapped data block numbers after applying the mapping function, wherein the mapped data block numbers are addresses of data of the file in a storage device. For example, NameNode determines “the addresses of the data nodes” (e.g., mapped data block numbers) on which the data block is stored. Data nodes (e.g., storage device(s)) store the data blocks of a file.

The client opens the file it wishes to read by calling `open()` on the `FileSystem` object, which for HDFS is an instance of `DistributedFileSystem` (step 1 in [Figure 3-2](#)). `DistributedFileSystem` calls the namenode, using remote procedure calls (RPCs), to determine the locations of the first few blocks in the file (step 2). For each block, the namenode returns the addresses of the datanodes that have a copy of that block. Furthermore, the datanodes are sorted according to their proximity to the client (according to the topology of the cluster's network; see “[Network Topology and Hadoop](#)” on [page 70](#)). If the client is itself a datanode (in the case of a MapReduce task, for instance), the client will read from the local datanode if that datanode hosts a copy of the block (see also [Figure 2-2](#) and “[Short-circuit local reads](#)” on [page 308](#)).



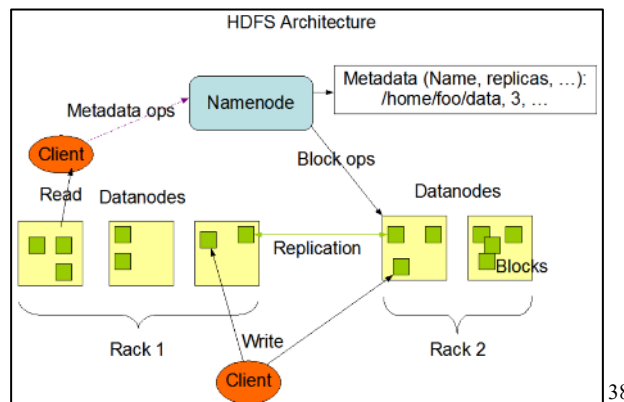
³⁶<https://ia800201.us.archive.org/2/items/HadoopForDummiesDirkDeRoos2014/Hadoop%20For%20Dummies%20-%20%20Dirk%20deRoos%202014.pdf>, at Page 75 of 411.

One important aspect of this design is that the client contacts datanodes directly to retrieve data and is guided by the namenode to the best datanode for each block. This design allows HDFS to scale to a large number of concurrent clients because the data traffic is spread across all the datanodes in the cluster. Meanwhile, the namenode merely has to service block location requests (which it stores in memory, making them very efficient) and does not, for example, serve data, which would quickly become a bottleneck as the number of clients grew.

37

NameNode and DataNodes

HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.



38

68. Defendant directly infringes the '686 Patent through its implementation and use of HDFS, performing at least the steps of claim 1.

69. Defendant has indirectly infringed and continues to indirectly infringe one or more claims of the '686 Patent by knowingly and intentionally inducing others, including NetApp customers and end-users, to directly infringe, either literally or under the doctrine of equivalents,

³⁷ <https://grut-computing.com/HadoopBook.pdf>, at Pages 97-98 of 756.

³⁸ <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>

by making, using, offering to sell, selling, and/or importing into the United States products that include the infringing technology.

70. Defendant, with knowledge that these products, or the use thereof, infringe the '686 Patent at least as of the date of this Complaint, knowingly and intentionally induced, and continues to knowingly and intentionally induce, direct infringement of the '686 Patent by providing these products to end-users for use in an infringing manner. Additionally, on information and belief, Defendant has adopted a policy of not reviewing the patents of others, and instructing its officers, agents, and employees to not review the patents of others, including specifically those related to Defendant's specific industry, with the subjective belief that there was a high probability that Defendant would learn of its infringing activities, thereby remaining willfully blind to the '686 Patent at least as early as the issuance of the '686 Patent

71. Defendant has induced infringement by others, including end-users, with the intent to cause infringing acts by others or, in the alternative, with the belief that there was a high probability that others, including end-users, infringe the '686 Patent, but while remaining willfully blind to the infringement. Defendant has and continues to induce infringement by their customers and end-users by supplying them with instructions on how to operate the infringing technology in an infringing manner, while also making publicly available information on the infringing technology via Defendant's website, product literature and packaging, and other publications..³⁹

72. Plaintiffs have suffered damages as a result of Defendant's direct and indirect infringement of the '686 Patent in an amount to be proven at trial.

³⁹ See, e.g., <https://www.netapp.com/media/16437-tr-4697.pdf>; see also the supporting proofs referenced herein.

73. Plaintiffs have suffered, and will continue to suffer, irreparable harm as a result of Defendant's infringement of the '686 Patent, for which there is no adequate remedy at law, unless Defendant's infringement is enjoined by this Court.

74. Defendant has committed and continues to commit acts of infringement that Defendant actually knew or should have known constituted an unjustifiably high risk of infringement of at least one valid and enforceable claim of the '686 Patent. Defendant's direct and indirect infringement of the '686 Patent has been and continues to be willful, intentional, deliberate, and/or in conscious disregard of rights under the patent. Plaintiffs are entitled to an award of treble damages, reasonable attorney fees, and costs in bringing this action.

COUNT IV
(Infringement of the '488 Patent)

75. Paragraphs 1 through 25 are incorporated by reference as if fully set forth herein.

76. Plaintiffs have not licensed or otherwise authorized Defendant to make, use, offer for sale, sell, or import any products that embody the inventions of the '488 Patent.

77. Defendant has infringed, and continues to directly infringe, the '488 Patent, either literally or under the doctrine of equivalents, without authority and in violation of 35 U.S.C. § 271, by making, using, offering to sell, selling, and/or importing into the United States products that satisfy each and every limitation of one or more claims of the '488 Patent. Such products include systems using NetApp ONTAP.

78. For example, Defendant has infringed, and continues to directly infringe, at least claim 1 of the '488 Patent by making, using, offering to sell, selling, and/or importing into the United States products that include NetApp ONTAP.

79. NetApp ONTAP provides a network, comprising a first electronic device associated with a first profile that uniquely identifies the first electronic device on the network.

For example, NetApp storage area networks (SAN) may be deployed on dedicated FAS or AFF nodes running ONTAP data management software. SAN nodes have adapters to connect to storage or to FC switches with optical cables. A Worldwide port name (WWPN) uniquely identifies a port on an adapter, and it is associated to a logical interface (LIF).

A storage area network (SAN) is a specialized, high-speed network that connects servers to storage devices, typically used by enterprises for business-critical applications that require high throughput and low latency.

40

How FC target nodes connect to the network

Storage systems and hosts have adapters, so they can be directly connected to each other or to FC switches with optical cables. For switch or storage system management, they might be connected to each other or to TCP/IP switches with Ethernet cables.

When a node is connected to the FC SAN, it registers each of its ports with the switch's Fabric Name Server service, using a unique identifier.

41

A LIF (logical interface) represents a network access point to a node in the cluster. You can configure LIFs on ports over which the cluster sends and receives communications over the network.

42

A world wide port name (WWPN) is a 64-bit value uniquely identifying a Fibre Channel port.

43

How FC nodes are identified

Each FC node is identified by a worldwide node name (WWNN) and a worldwide port name (WWPN).

44

⁴⁰ <https://www.netapp.com/data-storage/what-is-san-storage-area-network/>

⁴¹ <https://docs.netapp.com/a/ontap/7-mode/8.2.4/San-Administration-Guide-For-7-Mode.pdf>

⁴² https://docs.netapp.com/us-en/ontap/networking/configure_lifs_cluster_administrators_only_overview.html

⁴³ <https://docs.netapp.com/us-en/ontap-automation/resources/san.html>

⁴⁴ <https://docs.netapp.com/a/ontap/7-mode/8.2.4/San-Administration-Guide-For-7-Mode.pdf>

How WWPNs are used

WWPNs identify each port on an adapter. They are used for creating an initiator group and for uniquely identifying a storage system's HBA target ports.

- Creating an initiator group

The WWPNs of the host's HBAs are used to create an initiator group (igroup). An igroup is used to control host access to specific LUNs. You can create an igroup by specifying a collection of WWPNs of initiators in an FC network. When you map a LUN on a storage system to an igroup, you can grant all the initiators in that group access to that LUN. If a host's WWPN is not in an igroup that is mapped to a LUN, that host does not have access to the LUN. This means that the LUNs do not appear as disks on that host.

You can also create port sets to make a LUN visible only on specific target ports. A port set consists of a group of FC target ports. You can bind an igroup to a port set. Any host in the igroup can access the LUNs only by connecting to the target ports in the port set.

- Uniquely identifying a storage system's HBA target ports

The storage system's WWPNs uniquely identify each target port on the system. The host operating system uses the combination of the WWNN and WWPN to identify storage system adapters and host target IDs. Some operating systems require persistent binding to ensure that the LUN appears at the same target ID on the host.

45

What LIFs are

A LIF (logical interface) is an IP address or WWPN with associated characteristics, such as a role or a service policy, a home port, a home node, a list of ports to fail over to, and a firewall policy. You can configure LIFs on ports over which the cluster sends and receives communications over the network.

LIFs can be hosted on the following ports:

- Physical ports that are not part of interface groups
- Interface groups
- VLANs
- Physical ports or interface groups that host VLANs
- Virtual IP (VIP) ports

Starting with ONTAP 9.5, VIP LIFs are supported and are hosted on VIP ports.

While configuring SAN protocols such as FC on a LIF, it will be associated with a WWPN.

46

80. NetApp ONTAP comprises a second electronic device coupled to the first electronic device. For example, in a SAN network, nodes are interconnected (coupled) by using FC switches. SAN nodes have adapters with network ports to connect to storage or to FC switches by mean of optical cables.

⁴⁵ <https://docs.netapp.com/a/ontap/7-mode/8.2.4/San-Administration-Guide-For-7-Mode.pdf>

⁴⁶ <https://docs.netapp.com/p/activeiq/um/9.6/Online-Help.pdf>

What FC nodes are

In an FC network, nodes include targets, initiators, and switches.

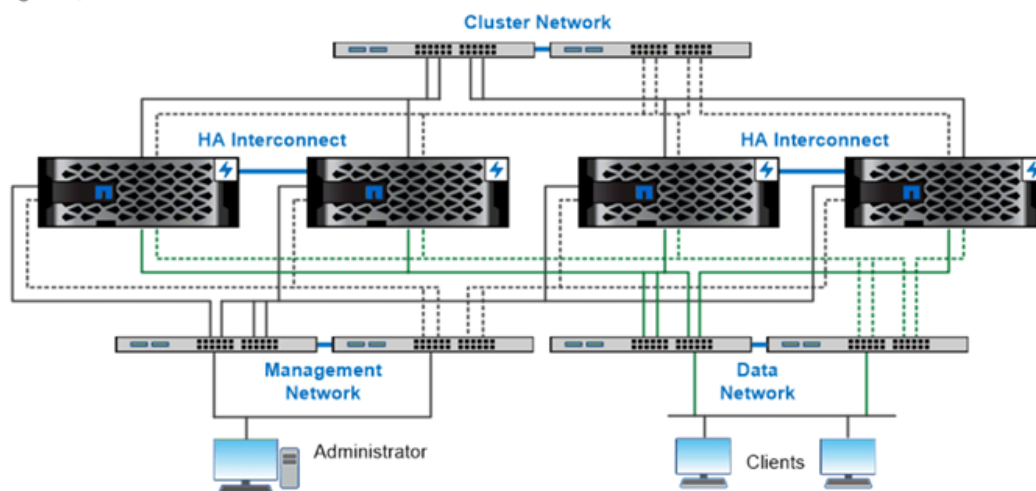
Targets are storage systems, and initiators are hosts. Nodes register with the Fabric Name Server when they are connected to an FC switch.

47

Network Architecture

The network architecture for an ONTAP data center implementation typically consists of a cluster interconnect, a management network for cluster administration, and a data network. Network interface cards (NICs) provide physical ports for Ethernet connections and host bus adapters (HBAs) provide physical ports for Fibre Channel (FC) connections.

Figure 1) ONTAP network architecture.



Logical and physical ports are provided on each node, you can use logical ports to manage network traffic. Logical ports are interface groups or virtual LANs (VLANs).

48

How FC target nodes connect to the network

Storage systems and hosts have adapters, so they can be directly connected to each other or to FC switches with optical cables. For switch or storage system management, they might be connected to each other or to TCP/IP switches with Ethernet cables.

When a node is connected to the FC SAN, it registers each of its ports with the switch's Fabric Name Server service, using a unique identifier.

49

81. NetApp ONTAP comprises a second electronic device coupled to the first electronic device; wherein the first profile is migrated to the second electronic device such that the

⁴⁷ <https://docs.netapp.com/a/ontap/7-mode/8.2.4/San-Administration-Guide-For-7-Mode.pdf>

⁴⁸ <https://www.netapp.com/media/17119-tr4760.pdf>

⁴⁹ <https://docs.netapp.com/a/ontap/7-mode/8.2.4/San-Administration-Guide-For-7-Mode.pdf>

first electronic device is no longer associated with the first profile and such that the first profile uniquely identifies the second electronic device on the network. For example, a SAN FC port on a LIF is associated with a WWPN, ONTAP can migrate a LIF to a different port on the SAN for load balancing purposes or as response to a failure. The SAN LIF configuration information, such as its WWPN and associated characteristics, is preserved when moving SAN LIFs to any node in a cluster.

You might have to migrate a LIF to a different port on the same node or a different node within the cluster, if the port is either faulty or requires maintenance. Migrating a LIF is similar to LIF failover, but LIF migration is a manual operation, while LIF failover is the automatic migration of a LIF in response to a link failure on the LIF's current network port.

50

If a node needs to be taken offline, you can move a SAN LIF to preserve its configuration information, such as its WWPN, and avoid rezoning the switch fabric. Because a SAN LIF must be taken offline before it is moved, host traffic must rely on host multipathing software to provide nondisruptive access to the LUN. You can move SAN LIFs to any node in a cluster, but you cannot move the SAN LIFs between storage virtual machines (SVMs).

The destination node and physical port for a LIF that you want to move must be on the same FC fabric or Ethernet network. If you move a LIF to a different fabric that has not been properly zoned, or if you move a LIF to an Ethernet network that does not have connectivity between iSCSI initiator and target, the LUN will be inaccessible when you bring it back online.

51

A world wide port name (WWPN) is a 64-bit value uniquely identifying a Fibre Channel port.

52

LIF failover refers to the automatic migration of a LIF to a different network port in response to a link failure on the LIF's current port. This is a key component to providing high availability for the connections to SVMs. Configuring LIF failover involves creating a failover group, modifying the LIF to use the failover group, and specifying a failover policy.

53

⁵⁰ https://docs.netapp.com/us-en/ontap/networking/migrate_a_lif.html

⁵¹ <https://docs.netapp.com/us-en/ontap/san-admin/move-san-lifs-task.html>

⁵² <https://docs.netapp.com/us-en/ontap-automation/resources/san.html>

⁵³ https://docs.netapp.com/us-en/ontap/networking/configure_failover_groups_and_policies_for_lifs_overview.html

You only need to perform a LIF movement if you are changing the contents of your cluster, for example, adding nodes to the cluster or deleting nodes from the cluster. If you perform a LIF movement, you do not have to re-zone your FC fabric or create new iSCSI sessions between the attached hosts of your cluster and the new target interface.

You cannot move a SAN LIF using the `network interface move` command. SAN LIF movement must be performed by taking the LIF offline, moving the LIF to a different home node or port, and then bringing it back online in its new location. Asymmetric Logical Unit Access (ALUA) provides redundant paths and automatic path selection as part of any ONTAP SAN solution. Therefore, there is no I/O interruption when the LIF is taken offline for the movement. The host simply retries and then moves I/O to another LIF.

54

You can revert a LIF to its home port after it fails over or is migrated to a different port either manually or automatically. If the home port of a particular LIF is unavailable, the LIF remains at its current port and is not reverted.

55

82. NetApp ONTAP comprises a second electronic device coupled to the first electronic device; wherein the first profile is migrated to the second electronic device such that the first electronic device is no longer associated with the first profile and such that the first profile uniquely identifies the second electronic device on the network; wherein said first profile comprises one or more identifiers selected from the group consisting of at least one Globally Unique Identifier (GUID), media access control (MAC) addresses, fibre channel world wide names, SAN boot parameters and a serial number associated with the first electronic device. For example, a SAN FC port on a LIF is associated with a WWPN identifier and its characteristics, such as a role or a service policy, a home port, a firewall policy, etc. ONTAP creates WWPN numbers based on serial numbers of host based adapters HBAs and MAC addresses when provisioning new Fibre Channel SAN LIFs.

⁵⁴ <https://docs.netapp.com/us-en/ontap/san-admin/lif-movement-concept.html>

⁵⁵ https://docs.netapp.com/us-en/ontap/networking/revert_a_lif_to_its_home_port.html

How storage systems are identified

When the FC protocol service is first initialized, it assigns a WWNN to a storage system based on the serial number of its NVRAM adapter. The WWNN is stored on disk.

Each target port on the HBAs installed in the storage system has a unique WWPN. Both the WWNN and the WWPN are a 64-bit address represented in the following format:

nn:nn:nn:nn:nn:nn:nn:nn, where n represents a hexadecimal value.

You can use commands such as `fcpl show adapter`, `fcpl config`, `sysconfig -v`, or `fcpl nodename` to see the system's WWNN as FC Nodename or nodename, or the system's WWPN as FC portname or portname.

56

If a node needs to be taken offline, you can move a SAN LIF to preserve its configuration information, such as its WWPN, and avoid rezoning the switch fabric. Because a SAN LIF must be taken offline before it is moved, host traffic must rely on host multipathing software to provide nondisruptive access to the LUN. You can move SAN LIFs to any node in a cluster, but you cannot move the SAN LIFs between storage virtual machines (SVMs).

The destination node and physical port for a LIF that you want to move must be on the same FC fabric or Ethernet network. If you move a LIF to a different fabric that has not been properly zoned, or if you move a LIF to an Ethernet network that does not have connectivity between iSCSI initiator and target, the LUN will be inaccessible when you bring it back online.

57

NetApp has added new MAC address Organizationally Unique Identifier (OUI) blocks.

- NetApp hardware is now being manufactured with network interface cards (NICs) with assigned MAC from a new address block.
- ONTAP uses the MAC of the network interface to calculate worldwide node names (WWNN) and worldwide port names (WWPN) when provisioning new Fibre Channel SAN logical interfaces (LIFs).
- Minimum ONTAP versions are required for storage controllers using the new MAC address block for SAN environments.
- Environments without SAN configurations will not be impacted by the MAC OUI change. However, NetApp encourages updating to the latest recommended release with any given release family to take advantage of additional bug fixes that may be available.

58

When a node is connected to the FC SAN, each SVM registers the World Wide Port Name (WWPN) of its LIF with the switch Fabric Name Service. The WWNN of the SVM and the WWPN of each LIF is automatically assigned by ONTAP.

59

⁵⁶ <https://docs.netapp.com/a/ontap/7-mode/8.2.4/San-Administration-Guide-For-7-Mode.pdf>

⁵⁷ <https://docs.netapp.com/us-en/ontap/san-admin/move-san-lifs-task.html>

⁵⁸ <https://kb.netapp.com/on-prem/ontap/OHW/OHW->

[KBs/NetApp_adds_new_MAC_address_OUI_range_to_shipping_systems](https://kb.netapp.com/on-prem/ontap/OHW/OHW-KBs/NetApp_adds_new_MAC_address_OUI_range_to_shipping_systems)

⁵⁹ <https://docs.netapp.com/us-en/ontap/san-admin/san-provisioning-fc-concept.html>

How worldwide name assignments work

Worldwide names are created sequentially in ONTAP. However, because of the way ONTAP assigns them, they might appear to be assigned in a non-sequential order.

Each adapter has a pre-configured WWPN and WWNN, but ONTAP does not use these pre-configured values. Instead, ONTAP assigns its own WWPNs or WWNNs, based on the MAC addresses of the onboard Ethernet ports.

The worldwide names might appear to be non-sequential when assigned for the following reasons:

- Worldwide names are assigned across all the nodes and storage virtual machines (SVMs) in the cluster.
- Freed worldwide names are recycled and added back to the pool of available names.

60

How FC nodes are identified

Each FC node is identified by a worldwide node name (WWNN) and a worldwide port name (WWPN).

61

83. Defendant directly infringes the '488 Patent through its implementation and use of NetApp ONTAP, performing at least the steps of claim 1.

84. Defendant has indirectly infringed and continues to indirectly infringe one or more claims of the '488 Patent by knowingly and intentionally inducing others, including NetApp customers and end-users, to directly infringe, either literally or under the doctrine of equivalents, by making, using, offering to sell, selling, and/or importing into the United States products that include the infringing technology.

85. Defendant, with knowledge that these products, or the use thereof, infringe the '488 Patent at least as of April 27, 2022,⁶² knowingly and intentionally induced, and continues to

⁶⁰ <https://docs.netapp.com/us-en/ontap/san-admin/san-provisioning-fc-concept.html#how-wwpns-are-used>

⁶¹ <https://docs.netapp.com/a/ontap/7-mode/8.2.4/San-Administration-Guide-For-7-Mode.pdf>

⁶² Plaintiffs sent Defendant correspondence on April 7, 2021, notifying them of their infringement of various patents, including some of the Asserted Patents. Thereafter, Plaintiffs and Defendant exchanged emails beginning in May 2021 regarding Defendant's infringement of various patents owned by Plaintiff. On April 27, 2022, Plaintiff sent Defendant an email with a claim chart identifying Defendant's infringement of the '488 Patent. Therefore, Defendant had knowledge of the '488 Patent at least as of April 2022.

knowingly and intentionally induce, direct infringement of the '488 Patent by providing these products to end-users for use in an infringing manner. Additionally, on information and belief, Defendant has adopted a policy of not reviewing the patents of others, and instructing its officers, agents, and employees to not review the patents of others, including specifically those related to Defendant's specific industry, with the subjective belief that there was a high probability that Defendant would learn of its infringing activities, thereby remaining willfully blind to the '488 Patent at least as early as the issuance of the '488 Patent

86. Defendant has induced infringement by others, including end-users, with the intent to cause infringing acts by others or, in the alternative, with the belief that there was a high probability that others, including end-users, infringe the '488 Patent, but while remaining willfully blind to the infringement. Defendant has and continues to induce infringement by their customers and end-users by supplying them with instructions on how to operate the infringing technology in an infringing manner, while also making publicly available information on the infringing technology via Defendant's website, product literature and packaging, and other publications..⁶³

87. Plaintiffs have suffered damages as a result of Defendant's direct and indirect infringement of the '488 Patent in an amount to be proven at trial.

88. Plaintiffs have suffered, and will continue to suffer, irreparable harm as a result of Defendant's infringement of the '488 Patent, for which there is no adequate remedy at law, unless Defendant's infringement is enjoined by this Court.

89. Defendant has committed and continues to commit acts of infringement that Defendant actually knew or should have known constituted an unjustifiably high risk of

⁶³ See, e.g., <https://www.netapp.com/data-management/ontap-data-management-software/>; see also the supporting proofs referenced herein.

infringement of at least one valid and enforceable claim of the '488 Patent. Defendant's direct and indirect infringement of the '488 Patent has been and continues to be willful, intentional, deliberate, and/or in conscious disregard of rights under the patent. Plaintiffs are entitled to an award of treble damages, reasonable attorney fees, and costs in bringing this action.

COUNT V
(Infringement of the '832 Patent)

90. Paragraphs 1 through 25 are incorporated by reference as if fully set forth herein.

91. Plaintiffs have not licensed or otherwise authorized Defendant to make, use, offer for sale, sell, or import any products that embody the inventions of the '832 Patent.

92. Defendant has directly infringed the '832 Patent, either literally or under the doctrine of equivalents, without authority and in violation of 35 U.S.C. § 271, by making, using, offering to sell, selling, and/or importing into the United States products that satisfy each and every limitation of one or more claims of the '832 Patent. Such products include NetApp SANtricity, as well as NetApp's systems that implement Apache Kafka Streams using RocksDB Universal Compaction, Apache Cassandra, and/or Apache HBase.

93. For example, Defendant has directly infringed at least claim 25 of the '832 Patent by making, using, offering to sell, selling, and/or importing into the United States products that include NetApp SANtricity, as well as NetApp's systems that implement Apache Kafka Streams using RocksDB Universal Compaction, Apache Cassandra, and/or Apache HBase.

NetApp SANtricity

94. NetApp SANtricity performs a method for monitoring performance of a storage device.

Abstract

With NetApp® Dynamic Disk Pools (DDP), storage administrators can group sets of like disks into a pool topology in which all the drives in the pool participate in the I/O workflow. This technology provides faster drive rebuilds than with RAID 5 or RAID 6 and removes the complications of RAID group configurations, so that storage administrators can focus on capacity allocation. This technical report provides a detailed description of the DDP feature.

Performance

DDP configurations are generally not the highest-performing configuration. However, with flash-based pools the delta when compared with RAID 5 and RAID 6 varies from insignificant with 100% read workloads to slightly less performance when running DDP with 100% write workloads. Figure 11 provides a general performance comparison using an EF570 array running 16KB and 64KB I/O as the workloads change from 100% write to 100% read.

Figure 16) Equal volumes in SANtricity System Manager.

Name	Status	Thin	Assigned To	LUN	Pool/Volume Group	Reported Capacity (GiB)	Edit
vol1	Optimal	No	Unassigned	None	Pool Disk_Pool_1	32288.00	
vol5	Optimal	No	Unassigned	None	Pool Disk_Pool_1	32288.00	
vol3	Optimal	No	Unassigned	None	Pool Disk_Pool_1	32288.00	
vol2	Optimal	No	Unassigned	None	Pool Disk_Pool_1	32288.00	
vol4	Optimal	No	Unassigned	None	Pool Disk_Pool_1	32288.00	

Total rows: 5

Figure 17) Pool capacity with equal volumes in SANtricity System Manager.

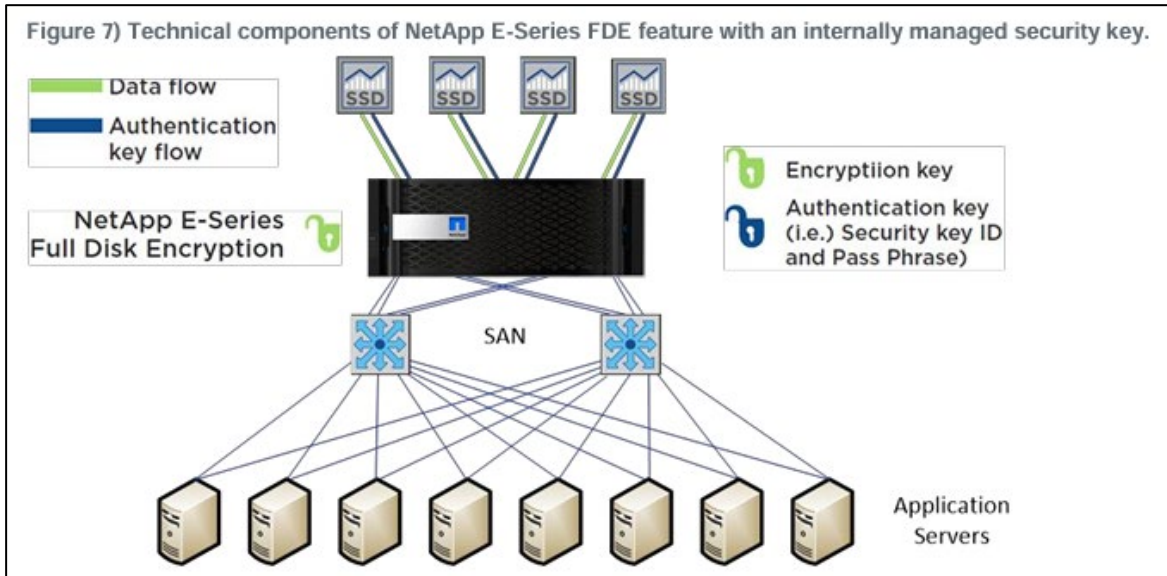
Name	Status	Total Drives	Capacity (GiB)	Allocated Capacity (GiB)	Free Capacity (GiB)	Total Volumes	Reported Volume Capacity (GiB)	Secured-Capable	Edit
Disk_Pool_1	Optimal	30	161456.00	161440.00	16.00	5	161440.00	No	

Total rows: 1

64

⁶⁴ <https://www.netapp.com/media/12421-tr-4652-santricity-os-ddp.pdf>.

95. NetApp SANtricity performs the step of intercepting communications between a computer system and said storage device.



- Predictable performance:
 - A deterministic algorithm dynamically distributes data, spare capacity, and protection information across a pool of drives.
 - If a drive fails, segments are recreated elsewhere, which reduces the magnitude and duration of the performance disruption.
 - The large pool of drives reduces hot spots.
 - Reduced exposure to multiple disk failures:
 - Through segment relocation, the system returns to an optimal state faster.
 - Through prioritized reconstruction, any stripes that experience multiple drive failures are given the highest priority.
- 66

96. NetApp SANtricity performs the step of analyzing said intercepted communications.

⁶⁵ <https://www.netapp.com/media/16851-tr-4623.pdf>

⁶⁶ <https://www.netapp.com/media/12421-tr-4652-santricity-os-ddp.pdf>

- Predictable performance:
 - A deterministic algorithm dynamically distributes data, spare capacity, and protection information across a pool of drives.
 - If a drive fails, segments are recreated elsewhere, which reduces the magnitude and duration of the performance disruption.
 - The large pool of drives reduces hot spots.
- Reduced exposure to multiple disk failures:
 - Through segment relocation, the system returns to an optimal state faster.
 - Through prioritized reconstruction, any stripes that experience multiple drive failures are given the highest priority.

67

97. NetApp SANtricity performs the step of reallocating at least some of said data on said storage device to enhance the performance of said storage device based on said analyzed communications.

A pool is a set of drives that is logically grouped. You can use a pool to create one or more volumes accessible to a host. (You create volumes from either a pool or a volume group.)

Pools can eliminate the need for administrators to monitor usage on each host to determine when they are likely to run out of storage space and avoid conventional disk resizing outages. When a pool nears depletion, additional drives can be added to the pool non-disruptively and capacity growth is transparent to the host.

With pools, data is automatically re-distributed to maintain equilibrium. By distributing parity information and spare capacity throughout the pool, every drive in the pool can be used to rebuild a failed drive. This approach does not use dedicated hot spare drives; instead, preservation (spare) capacity is reserved throughout the pool. Upon drive failure, segments on other drives are read to recreate the data. A new drive is then chosen to write each segment that was on a failed drive so that data distribution across drives is maintained.

68

Apache Kafka Streams using RocksDB Universal Compaction

98. On information and belief, NetApp has used Apache Kafka Streams using RocksDB Universal Compaction in the U.S. since at least 2022. The default state store type of Apache Kafka Streams is RocksDB. Apache Kafka Streams also uses universal compaction by default for its state stores.

⁶⁷ <https://www.netapp.com/media/12421-tr-4652-santricity-os-ddp.pdf>

⁶⁸ <https://docs.netapp.com/us-en/e-series-santricity/sm-storage/what-is-a-pool.html>

NetApp Closes Acquisition of Instacluster

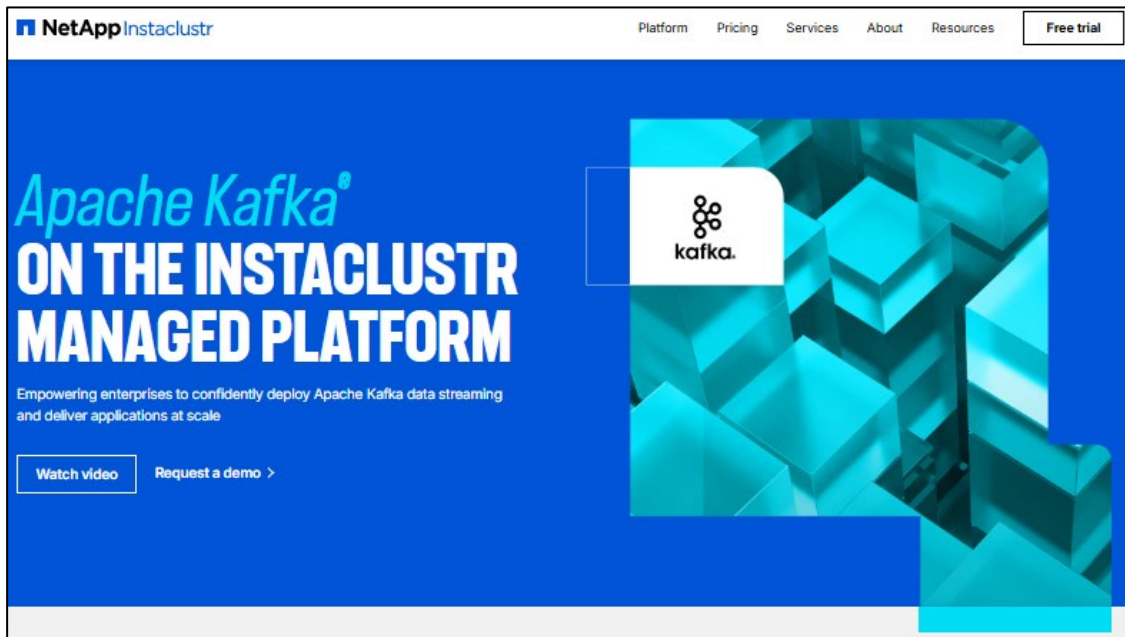
SAN JOSE, Calif.-(BUSINESS WIRE)--May 24, 2022-- NetApp® (NASDAQ: NTAP), a global, cloud-led, data-centric software company, today announced it has completed its acquisition of Instacluster, a leading platform provider of fully managed open-source database, pipeline and workflow applications delivered as a service.

69

Companies producing software are becoming ever more reliant on open source databases to build their programs, but it's complex working with all these different products. [Instacluster](#), a California startup, wanted to change that by offering popular open source databases delivered as a service.

Today, NetApp bought the company for an undisclosed amount. Instacluster gives NetApp customers a way to install popular open source databases without worrying about the headaches of dealing with the raw open source. Among their supported projects are PostgreSQL, Apache Kafka, OpenSearch, ElasticSearch and Apache Cassandra.

70



71

⁶⁹ <https://investors.netapp.com/news-releases/news-release-details/netapp-closes-acquisition-instacluster>

⁷⁰ <https://techcrunch.com/2022/04/07/netapp-acquires-instacluster-to-deliver-open-source-databases-as-a-service/>

⁷¹ <https://www.instacluster.com/platform/managed-apache-kafka/>

Available on-prem and cloud platforms

aws Microsoft Azure Google Cloud

WHAT IS Apache Kafka?

Apache Kafka is an open source distributed streaming platform designed to handle high-volume, real-time data streams. It provides a scalable and fault-tolerant architecture for ingesting, storing, and processing data in a publish-subscribe model.

Kafka's key features include high throughput, low latency, fault tolerance, and the ability to handle massive data streams across multiple applications and systems, making it a popular choice for building real-time data pipelines and event-driven architectures.

72

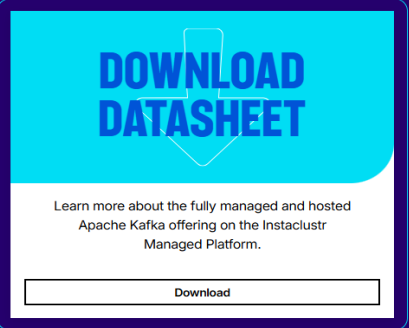
Platform Intelligent, open source application data infrastructure > Explore our platform Security and trust Enterprise-grade security > Learn more Hosting aws Microsoft Azure Google Cloud Data infrastructure management in the cloud and on-prem > Learn more	Store PostgreSQL® Apache Cassandra® Valkey™ Orchestrate Cadence Apache ZooKeeper™ Search OpenSearch	Stream Apache Kafka® Kafka® Connect Analyze ClickHouse®
---	---	---

73

⁷² <https://www.instaclustr.com/platform/managed-apache-kafka/>

⁷³ <https://www.instaclustr.com/>

MEET INSTACLUSTER FOR *Apache Kafka*



Learn more about the fully managed and hosted Apache Kafka offering on the Instaclustr Managed Platform.

Download

Instaclustr for Apache Kafka runs Kafka in the cloud or on-prem, providing a production-ready and fully supported Kafka clusters in minutes.

[Discover our platform](#)

Fully managed and hosted, 100% open source

- Run in Your Cloud Provider Account or Ours
- 24x7 Support
- SOC2, ISO27001 and ISO27018 Certified
- PCI-DSS and HIPAA Compliant
- Built in Monitoring
- Provision via Console, API or Terraform Provider

99.999% SLA	+
Dedicated Apache ZooKeeper™ or KRaft nodes	+
Easily add Kafka® Connect	+
Managed mirroring	+
Zero Downtime Migration	+

74

NetApp Instaclustr

Platform Datasheet

APACHE KAFKA®

Fully hosted and managed service



More than 80% of the Fortune 500 rely on Apache Kafka® for message streaming and queueing. Kafka is second to none in its ability to support robust, event-driven architectures. But it's also a sophisticated piece of technology that can require major investment to manage effectively.

With NetApp® Instaclustr for Apache Kafka, you don't have to hire multiple engineers or spend months planning a deployment. Instead, you can get a production-ready Kafka cluster with the click of a button, backed by our industry-leading SLAs and expert Support team. Offload managing Kafka to us so you can focus on innovation in the rest of your application stack.

75

⁷⁴ <https://www.instaclustr.com/platform/managed-apache-kafka/>

⁷⁵ https://www.instaclustr.com/wp-content/uploads/Insta_Platform_Datasheet_Apache_Kafka_1_27aug24.pdf

Features • 100% Open source Apache Kafka • Easy-to-use management console • Core software bug fixes included • Supported software bug fixes included • Repair coordination built-in • Managed mirroring • Dedicated ZooKeeper/KRaft nodes • Single-tenant clusters by default • Terraform-based provisioning • Prometheus monitoring API • Provisioning and monitoring RESTful APIs • Graviton2, Graviton3, and Intel nodes available	Available add-ons • Managed Apache Kafka® Connect • Managed Apache Kafka MirrorMaker 2 • Apache Kafka/Karapace • AWS PrivateLink • GCP Private Service Connect • Private Network Cluster (no publicly routable IPs for nodes)
SLAs • Up to 99.999% availability • Read/write latency	Flexible hosting options • AWS • Microsoft Azure • Google Cloud • Digital Ocean • IBM Cloud • On-prem
Support • 24x7 expert coverage included • Guaranteed 15 minute response time, 24x7 work to resolve critical issues	Security and compliance • SOC 2 certified • PCI-DSS available • HIPAA available • GDPR • ISO 27001 and ISO 27018 certified

76

1. Countdown begins: The week before ApacheCon

I'm looking forward to [ApacheCon 2019](#) next week in Las Vegas. The Apache Software Foundation has been around since 1999 when it started out with a single project, their "Apache" HTTP Server. Fast forward to 2019 (20 years on) and there are 334 projects and 48 incubating! I'm expecting the conference to be abuzz with this level of activity on display. Three of the Open Source technologies that Instaclustr provides as managed services are Apache projects, [Cassandra](#), [Kafka](#), and [Spark](#), so ApacheCon is an obvious choice of destination for [Instaclustr to attend](#) (as sponsors) and speak at this year. The conference spreads over 4 days next week with a great choice of presentations in multiple tracks including: Cassandra, IoT, Geospatial, Streaming, Machine Learning, and Observability! We have talks in the Cassandra and Observability tracks, but I'm attending (and blogging about) talks in other tracks that have interesting or challenging use cases (e.g. IoT, geospatial, ML, streaming, etc.), technologies that are complementary to Cassandra and Kafka, or that are directly competing to better understand the technical and business tradeoffs for our customers.

77

⁷⁶ https://www.instaclustr.com/wp-content/uploads/Insta_Platform_Datasheet_Apache_Kafka_1_27aug24.pdf

⁷⁷ <https://www.instaclustr.com/blog/apachecon-2019-las-vegas-report-core-complementary-and-competing-technologies-and-keynotes/>

WHY CHOOSE INSTACLUSTER FOR OPEN SOURCE?

 Worldwide team We have one of the largest, most experienced teams of Cassandra engineers in the industry, with more than 30 dedicated Cassandra support experts across the globe	 Comprehensive data platform Seamlessly host and manage Apache Cassandra®, Apache Kafka®, Apache Spark®, ZooKeeper™, Valkey, OpenSearch®, PostgreSQL®, and Cadence®
 Expert support Support for the latest Cassandra features and versions (including 5.0!) on our managed platform	 Source code experts Collaborative expertise in Cassandra with a dedicated committer on staff
 Experienced management More than 360 million node hours of experience managing open source technologies including Cassandra	 Seamless migration with zero downtime From proprietary (e.g., DataStax, DynamoDB) or self-managed Cassandra clusters
 100% Open source We'll never charge you license fees or lock you in with complicated contracts	

78

WHY CHOOSE INSTACLUSTER FOR OPEN SOURCE?

 Worldwide team Largest and most experienced team of open source operators in the industry	 Seamless migration with zero downtime From proprietary (e.g. Confluent) or self-managed Kafka clusters
 Experienced management More than 360 million node hours of experience managing open source technologies including Kafka	 Source code experts Collaborative expertise in Kafka with a dedicated committer on staff
 100% Open source We'll never charge you license fees or lock you in with hidden, proprietary features	 Comprehensive data platform Seamlessly host and manage Apache Cassandra®, Apache ZooKeeper™, Apache Kafka® Connect, OpenSearch®, Apache Spark™, Valkey, PostgreSQL®, ClickHouse, and Cadence®

79

⁷⁸ https://www.instacluster.com/wp-content/uploads/2023/03/Instacluster_Cassandra_Managed_Service_Datasheet.pdf

⁷⁹ https://www.instacluster.com/wp-content/uploads/Insta_Platform_Datasheet_Apache_Kafka_1_27aug24.pdf



Sr Kafka Admin/Engineer

Instaclustr · Contract

Mar 2021 - Feb 2022 · 1 yr

San Francisco Bay Area

- Installed the Apache Kafka cluster and Confluent Kafka open source in different environments.
- Maintaining Atlassian products like JIRA, Confluence, Bamboo, and Bitbucket.
- Worked with Linux/UNIX environments and scripting for Build & Release automation.
- Supported AWS Cloud environment with 500+ AWS instances and configured Elastic IP Elastic Storage and also experience working on implemented security groups, Followed by automation of continuous integration and continuous delivery process (CI/CD) of build and release
- management using agile methodologies, Which include monitoring, configuration troubleshooting, and • maintaining cloud and DevOps environment with the required toolset.
- Prepared test scenarios using Gherkin language as part of the BDD approach.
- Implemented real-time log analytics pipeline using Confluent Kafka, Storm, and elastic search Logstash kibana.

80

default.dsl.store

The default state store type used by DSL operators.

Type: string
Default: rocksDB
Valid Values: [rocksDB, in_memory]
Importance: low

81

Use level compaction

Kafka Streams uses universal compaction by default for its state stores. However, level compaction has lower space amplification than universal compaction at the cost of lower write rates. The corresponding option that you can tune in your `RocksDBConfigSetter` implementation is `options.setCompactionStyle(CompactionStyle.LEVEL)`.

82

99. Apache Kafka Streams using RocksDB Universal Compaction performs a method for monitoring performance of a storage device.

⁸⁰ <https://www.linkedin.com/in/bharat-katragadda-2b663519a/>

⁸¹ <https://kafka.apache.org/documentation/#introduction>

⁸² <https://www.confluent.io/blog/how-to-tune-rocksdb-kafka-streams-state-stores-performance/>

Optimized for Fast Storage

RocksDB is optimized for fast, low latency storage such as flash drives and high-speed disk drives. RocksDB exploits the full potential of high read/write rates offered by flash or RAM.

RocksDB provides several APIs to read KV pairs from a database, including Get and MultiGet for point lookups and Iterator for sequential scanning. These APIs may result in RocksDB reading blocks from SST files on disk storage. The types of blocks and the frequency with which they are read from storage is workload dependent. Some workloads may have a small working set and thus may be able to cache most of the data required, while others may have large working sets and have to read from disk more often. In the latter case, the latency would be much higher and throughput would be lower than the former. They would also be dependent on the characteristics of the underlying storage media, making it difficult to migrate from one medium to another, for example, local flash to disaggregated flash.

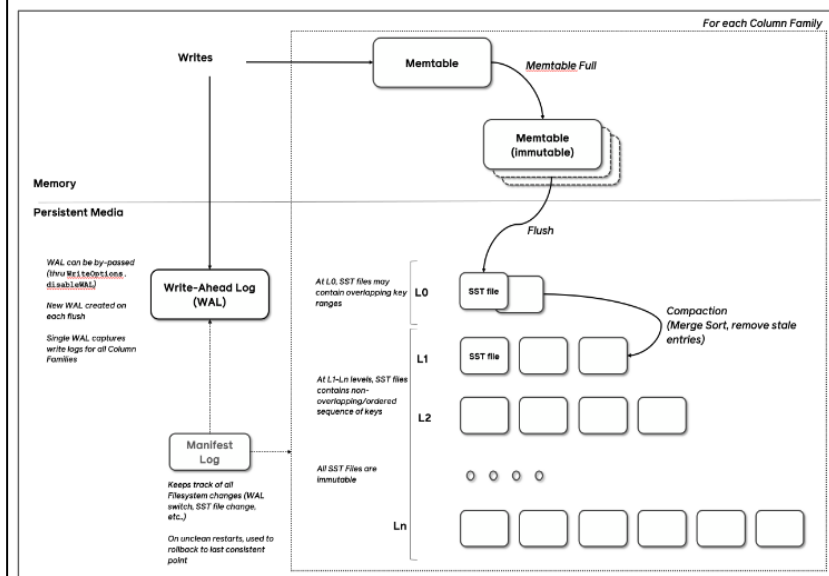
83

3. High Level Architecture

RocksDB is a storage engine library of key-value store interface where keys and values are arbitrary byte streams. RocksDB organizes all data in sorted order and the common operations are `Get(key)`, `NewIterator()`, `Put(key, val)`, `Delete(key)`, and `SingleDelete(key)`.

The three basic constructs of RocksDB are **memtable**, **sstfile** and **logfile**. The **memtable** is an in-memory data structure - new writes are inserted into the **memtable** and are optionally written to the **logfile** (aka. **Write Ahead Log(WAL)**). The logfile is a sequentially-written file on storage. When the memtable fills up, it is flushed to a **sstfile** on storage and the corresponding logfile can be safely deleted. The data in an **sstfile** is sorted to facilitate easy lookup of keys.

The default format of **sstfile** is described in more details [here](#).



84

⁸³ <https://rocksdb.org/blog/>

⁸⁴ <https://github.com/facebook/rocksdb/wiki/RocksDB-Overview>

Kafka Streams uses universal compaction because it supports high write rates. In universal compaction, let's say we have five SST files with a total size x . When we compact from one level to another, we write no more than the size of the input files, so the write amplification of the compaction job is $1x$. In leveled compaction, a set of files of size x in L1 gets compacted with files in L2 that are usually 10 times bigger, so the write amplification of a compaction job would be $11x$.

Use level compaction

Kafka Streams uses universal compaction by default for its state stores. However, level compaction has lower space amplification than universal compaction at the cost of lower write rates. The corresponding option that you can tune in your RocksDBConfigSetter implementation is `options.setCompactionStyle(CompactionStyle.LEVEL)`.

85

Compaction Styles

Both Level Style Compaction and Universal Style Compaction store data in a fixed number of logical *levels* in the database. More recent data is stored in Level-0 (L0) and older data in higher-numbered levels, up to L_{max} . Files in L0 may have overlapping keys, but files in other levels generally form a single sorted run per level.

Level Style Compaction (default) typically optimizes disk footprint vs. logical database size (space amplification) by minimizing the files involved in each compaction step: merging one file in L_n with all its overlapping files in L_{n+1} and replacing them with new files in L_{n+1} .

Universal Style Compaction typically optimizes total bytes written to disk vs. logical database size (write amplification) by merging potentially many files and levels at once, requiring more temporary space. Universal typically results in lower write-amplification but higher space- and read-amplification than Level Style Compaction.

86

100. Apache Kafka Streams using RocksDB Universal Compaction performs the step of intercepting communications between a computer system and said storage device.

⁸⁵ <https://www.confluent.io/blog/how-to-tune-rocksdb-kafka-streams-state-stores-performance/>

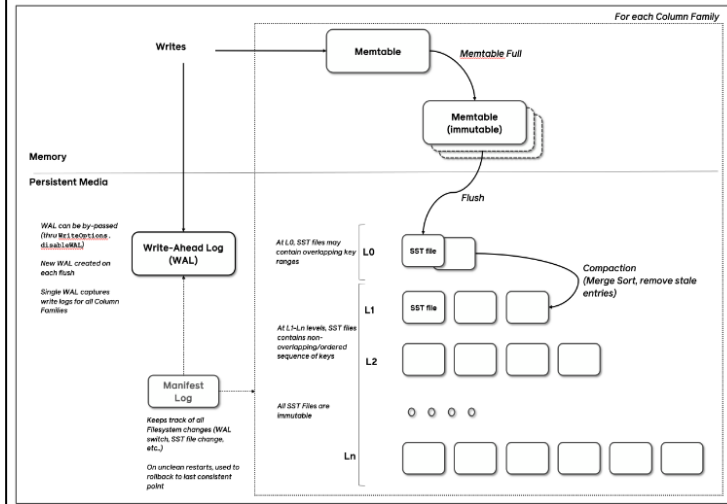
⁸⁶ <https://github.com/facebook/rocksdb/wiki/RocksDB-Overview>

3. High Level Architecture

RocksDB is a storage engine library of key-value store interface where keys and values are arbitrary byte streams. RocksDB organizes all data in sorted order and the common operations are `Get(key)`, `NewIterator()`, `Put(key, val)`, `Delete(key)`, and `SingleDelete(key)`.

The three basic constructs of RocksDB are **memtable**, **sstfile** and **logfile**. The **memtable** is an in-memory data structure - new writes are inserted into the **memtable** and are optionally written to the **logfile** (aka. **Write Ahead Log(WAL)**). The logfile is a sequentially-written file on storage. When the memtable fills up, it is flushed to a **sstfile** on storage and the corresponding logfile can be safely deleted. The data in an sstfile is sorted to facilitate easy lookup of keys.

The default format of sstfile is described in more details [here](#).



87

101. Apache Kafka Streams using RocksDB Universal Compaction performs the step of analyzing said intercepted communications.

3. High Level Architecture

RocksDB is a storage engine library of key-value store interface where keys and values are arbitrary byte streams. RocksDB organizes all data in sorted order and the common operations are `Get(key)`, `NewIterator()`, `Put(key, val)`, `Delete(key)`, and `SingleDelete(key)`.

The three basic constructs of RocksDB are **memtable**, **sstfile** and **logfile**. The **memtable** is an in-memory data structure - new writes are inserted into the **memtable** and are optionally written to the **logfile** (aka. **Write Ahead Log(WAL)**). The logfile is a sequentially-written file on storage. When the memtable fills up, it is flushed to a **sstfile** on storage and the corresponding logfile can be safely deleted. The data in an sstfile is sorted to facilitate easy lookup of keys.

88

Precondition: $n \geq \text{options.level0_file_num_compaction_trigger}$

Unless number of sorted runs reaches this threshold, no compaction will be triggered at all.

⁸⁷ <https://github.com/facebook/rocksdb/wiki/RocksDB-Overview>

⁸⁸ <https://github.com/facebook/rocksdb/wiki/Universal-Compaction>

If the pre-condition is satisfied, there are four conditions. Each of them can trigger a compaction. They are evaluated in order and the selection stops once a compaction has been scheduled. Before going into detail, the overview for picking a compaction is:

1. Try to schedule a compaction by age of data
2. Else if the space amplification constraint has been violated, try to schedule a major compaction
3. Else try to schedule a compaction (minor or major) while respecting `size_ratio`
4. Else try to schedule a minor compaction without respecting `size_ratio`

89

1. Compaction triggered by age of data

For universal style compaction, the aging-based triggering criterion has the highest priority since it is a hard requirement. When trying to pick a compaction, this condition is checked first.

If the condition meets (there are files older than `options.periodic_compaction_seconds`), then RocksDB proceeds to pick sorted runs for compaction. RocksDB picks sorted runs from oldest to youngest until encountering a sorted run that is being compacted by another compaction. These files will be compacted to bottommost level unless bottommost level is reserved for ingestion behind. In this case, files will be compacted to second bottommost level.

2. Compaction Triggered by Space Amplification

If the estimated *size amplification ratio* is larger than `options.compaction_options_universal.max_size_amplification_percent / 100`, all files will be compacted to one sorted run.

3. Compaction Triggered by number of sorted runs while respecting `size_ratio`

If a compaction is not triggered by the space amplification constraint as described above then a compaction can be triggered by the number of sorted runs. This step respects *size ratio trigger* when selecting the sorted runs to compact.

We calculated a value of *size ratio trigger* as

```
size_ratio_trigger = (100 + options.compaction_options_universal.size_ratio) / 100
```

4. Compaction Triggered by number of sorted runs without respecting `size_ratio`

If none of the previous conditions cause a compaction to be scheduled, then this condition might schedule one. This step does not respect *size ratio trigger* but the logic is otherwise similar to what was described above for condition 3. If we need to compact more than `options.compaction_options_universal.max_merge_width` number of sorted runs, we cap it to `options.compaction_options_universal.max_merge_width`.

90

102. Apache Kafka Streams using RocksDB Universal Compaction performs the step of reallocating at least some of said data on said storage device to enhance the performance of said storage device based on said analyzed communications.

⁸⁹ <https://github.com/facebook/rocksdb/wiki/Terminology>

⁹⁰ <https://github.com/facebook/rocksdb/wiki/Universal-Compaction>

In the presence of ongoing writes, compactions are needed for space efficiency, read (query) efficiency, and timely data deletion. Compaction removes key-value bindings that have been deleted or overwritten, and re-organizes data for query efficiency. Compactions may occur in multiple threads if configured.

The entire database is stored in a set of *sstfiles*. When a *memtable* is full, its content is written out to a file in Level-0 (L0) of the LSM tree. RocksDB removes duplicate and overwritten keys in the memtable when it is flushed to a file in L0. In *compaction*, some files are periodically read in and merged to form larger files, often going into the next LSM level (such as L1, up to Lmax).

The overall write throughput of an LSM database directly depends on the speed at which compactions can occur, especially when the data is stored in fast storage like SSD or RAM. RocksDB may be configured to issue concurrent compaction requests from multiple threads. It is observed that sustained write rates may increase by as much as a factor of 10 with multi-threaded compaction when the database is on SSDs, as compared to single-threaded compactions.

91

Apache Cassandra

103. On information and belief, NetApp has used Apache Cassandra in the U.S. since at least 2022.

NetApp Closes Acquisition of Instaclustr

SAN JOSE, Calif. --(BUSINESS WIRE)--May 24, 2022-- NetApp® (NASDAQ: NTAP), a global, cloud-led, data-centric software company, today announced it has completed its acquisition of Instaclustr, a leading platform provider of fully managed open-source database, pipeline and workflow applications delivered as a service.

92

On May 20, 2022, in the first quarter of fiscal 2023, we acquired all the outstanding shares of privately-held Instaclustr, Inc., a leading platform provider of fully managed open-source database, pipeline and workflow applications delivered as a service, for approximately \$500 million.

93

Companies producing software are becoming ever more reliant on open source databases to build their programs, but it's complex working with all these different products. [Instaclustr](#), a California startup, wanted to change that by offering popular open source databases delivered as a service.

Today, NetApp bought the company for an undisclosed amount. Instaclustr gives NetApp customers a way to install popular open source databases without worrying about the headaches of dealing with the raw open source. Among their supported projects are PostgreSQL, Apache Kafka, OpenSearch, Elasticsearch and Apache Cassandra.

94

⁹¹ <https://github.com/facebook/rocksdb/wiki/RocksDB-Overview>

⁹² <https://investors.netapp.com/news-releases/news-release-details/netapp-closes-acquisition-instaclustr>

⁹³ See NetApp Form 10-K for Fiscal Year 2022, available at: <https://investors.netapp.com/static-files/d5e272be-842a-49b4-8871-93469a68af0d>

⁹⁴ <https://techcrunch.com/2022/04/07/netapp-acquires-instaclustr-to-deliver-open-source-databases-as-a-service/>

NetApp Instacluster Platform Pricing Services About Resources **Free trial**

UNLOCK PEAK PERFORMANCE

For open source technologies

Simplify data infrastructure management to effortlessly scale and enhance the performance of your applications both on-prem and across clouds.

95

Platform Intelligent, open source application data infrastructure > Explore our platform Security and trust Enterprise-grade security > Learn more Hosting aws, Microsoft, Google, Oracle Data infrastructure management in the cloud and on-prem > Learn more	Store PostgreSQL® Apache Cassandra® Valkey™ Orchestrate Cadence Apache ZooKeeper™ Search OpenSearch	Stream Apache Kafka® Kafka® Connect Analyze ClickHouse®
--	---	--

96

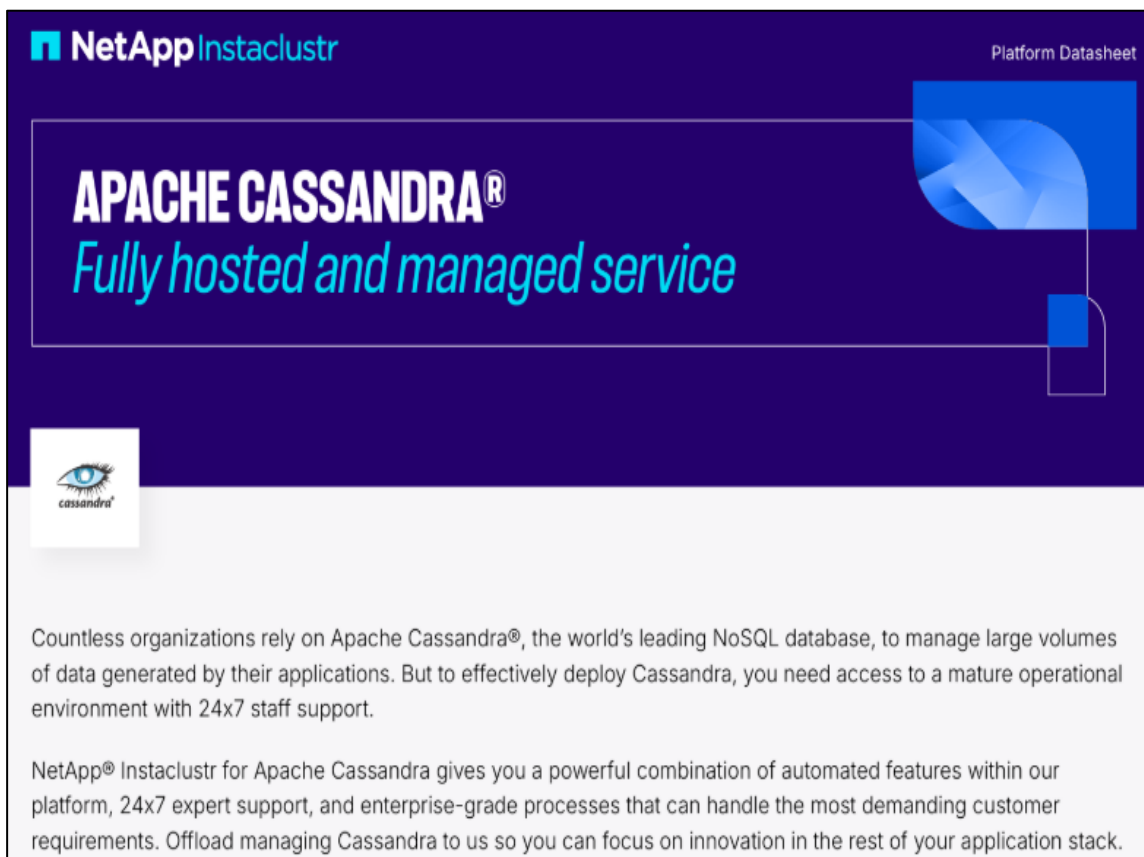
⁹⁵ <https://www.instacluster.com/>

⁹⁶ <https://www.instacluster.com/>

1. Countdown begins: The week before ApacheCon

I'm looking forward to [ApacheCon 2019](#) next week in Las Vegas. The Apache Software Foundation has been around since 1999 when it started out with a single project, their "Apache" HTTP Server. Fast forward to 2019 (20 years on) and there are 334 projects and 48 incubating! I'm expecting the conference to be abuzz with this level of activity on display. Three of the Open Source technologies that InstaClustr provides as managed services are Apache projects, [Cassandra](#), [Kafka](#), and [Spark](#), so ApacheCon is an obvious choice of destination for [InstaClustr to attend](#) (as sponsors) and speak at this year. The conference spreads over 4 days next week with a great choice of presentations in multiple tracks including: Cassandra, IoT, Geospatial, Streaming, Machine Learning, and Observability! We have talks in the Cassandra and Observability tracks, but I'm attending (and blogging about) talks in other tracks that have interesting or challenging use cases (e.g. IoT, geospatial, ML, streaming, etc.), technologies that are complementary to Cassandra and Kafka, or that are directly competing to better understand the technical and business tradeoffs for our customers.

97



NetApp InstaClustr Platform Datasheet

APACHE CASSANDRA®
Fully hosted and managed service



Countless organizations rely on Apache Cassandra®, the world's leading NoSQL database, to manage large volumes of data generated by their applications. But to effectively deploy Cassandra, you need access to a mature operational environment with 24x7 staff support.

NetApp® InstaClustr for Apache Cassandra gives you a powerful combination of automated features within our platform, 24x7 expert support, and enterprise-grade processes that can handle the most demanding customer requirements. Offload managing Cassandra to us so you can focus on innovation in the rest of your application stack.

98

⁹⁷ <https://www.instaclustr.com/blog/apachecon-2019-las-vegas-report-core-complementary-and-competing-technologies-and-keynotes/>

⁹⁸ https://www.instaclustr.com/wp-content/uploads/2023/03/Instaclustr_Cassandra_Managed_Service_Datasheet.pdf

WHY CHOOSE INSTACLUSTER FOR OPEN SOURCE?

	Worldwide team We have one of the largest, most experienced teams of Cassandra engineers in the industry, with more than 30 dedicated Cassandra support experts across the globe		Comprehensive data platform Seamlessly host and manage Apache Cassandra®, Apache Kafka®, Apache Spark®, ZooKeeper™, Valkey, OpenSearch®, PostgreSQL®, and Cadence®
	Expert support Support for the latest Cassandra features and versions (including 5.0!) on our managed platform		Source code experts Collaborative expertise in Cassandra with a dedicated committer on staff
	Experienced management More than 360 million node hours of experience managing open source technologies including Cassandra		Seamless migration with zero downtime From proprietary (e.g., DataStax, DynamoDB) or self-managed Cassandra clusters
	100% Open source We'll never charge you license fees or lock you in with complicated contracts		

99

⁹⁹ https://www.instaclustr.com/wp-content/uploads/2023/03/Instaclustr_Cassandra_Managed_Service_Datasheet.pdf

Features

- 100% Open source Apache Cassandra
- Easy-to-use management console
- Core software bug fixes included
- Supported software bug fixes included
- Repair coordination built-in
- Ongoing backups built-in
- Single-tenant clusters by default
- Multi-region clusters available
- Terraform-based provisioning
- Prometheus monitoring API
- Provisioning and monitoring RESTful APIs
- ecAudit support
- LDAP and Kerberos authentication
- Self-service horizontal and vertical scaling
- AWS PrivateLink

Available add-ons

- Debezium® Change Data Capture Connector for Managed Apache Cassandra
- Private Network Cluster (no publicly routable IPs for nodes)
- Continuous Backups (snapshot backups every 3 hours, commit log every 5 minutes)

Flexible hosting options

- AWS
- Microsoft Azure
- Google Cloud
- On-prem

Security and compliance

- SOC 2 Certified
- PCI-DSS available
- HIPAA available
- ISO 27001 and ISO 27018 Certified

SLAs

- Up to 100% availability
- Read/write latency

Support

- 24x7 expert coverage included
- Guaranteed 15 minute response time, 24x7 work to resolve critical issues

100

104. Apache Cassandra performs a method for monitoring performance of a storage device. For example, Apache Cassandra monitors performance metrics, such as through “ReadLatency.” By way of further example, Apache Cassandra performs compaction, which results in reclaiming disk space.

ReadLatency	Latency	Local read latency for this table.
-------------	---------	------------------------------------

101

¹⁰⁰ https://www.instaclustr.com/wp-content/uploads/2023/03/Instaclustr_Cassandra_Managed_Service_Datasheet.pdf

¹⁰¹ <https://cassandra.apache.org/doc/latest/cassandra/operating/metrics.html>

Cassandra Compaction is a process of reconciling various copies of data spread across distinct SSTables. Cassandra performs compaction of SSTables as a background activity. Cassandra has to maintain fewer SSTables and fewer copies of each data row due to compactions improving its read performance. Compaction is a crucial area of Cassandra performance and maintenance.

Cassandra's Write Path

To understand the importance of compactions in Cassandra, you must first understand how Cassandra writes data to a disk. The Cassandra write path in a nutshell:

1. Cassandra stores recent writes in memory (in a structure called the Memtable).
2. When enough writes have been made, Cassandra flushes the Memtable to disk. Data on disk is stored in relatively simple data structures called Sorted String Tables (SSTable). At the most simplified level, an SSTable could be described as a sorted array of strings.
3. Before writing a new SSTable, Cassandra merges and pre-sorts the data in the Memtable according to Primary Key. In Cassandra, a Primary Key consists of a Partition Key (the unique key that determines which node the data is stored on) and any Clustering Keys that have been defined.
4. The SSTable is written to disk as a single contiguous write operation. SSTables are immutable. Once they are written to disk they are not modified. Any updates to data or deletion of data within an SSTable is written to a new SSTable. If data is updated regularly, Cassandra may need to read from multiple SSTables to retrieve a single row.
5. Compaction operations occur periodically to re-write and combine SSTables. This is required because SSTables are immutable (no modifications once written to disk). Compactions prune deleted data and merge disparate row data into new SSTables in order to reclaim disk space and keep read operations optimized.

If you are unfamiliar with Cassandra's write path, please read [The write path to compaction](#) from Cassandra Wiki.

102

¹⁰² <https://www.instaclustr.com/blog/apache-cassandra-compaction/>

Compaction

Strategies

Picking the right compaction strategy for your workload will ensure the best performance for both querying and for compaction itself.

Size Tiered Compaction Strategy (STCS)

The default compaction strategy. Useful as a fallback when other strategies don't fit the workload. Most useful for non pure time series workloads with spinning disks, or when the I/O from `LCS` is too high.

Leveled Compaction Strategy (LCS)

Leveled Compaction Strategy (LCS) is optimized for read heavy workloads, or workloads with lots of updates and deletes. It is not a good choice for immutable time series data.

Time Window Compaction Strategy (TWCS)

Time Window Compaction Strategy is designed for TTL'ed, mostly immutable time series data.

Types of compaction

The concept of compaction is used for different kinds of operations in Cassandra, the common thing about these operations is that it takes one or more SSTables and output new SSTables. The types of compactions are:

Minor compaction

triggered automatically in Cassandra.

Major compaction

a user executes a compaction over all SSTables on the node.

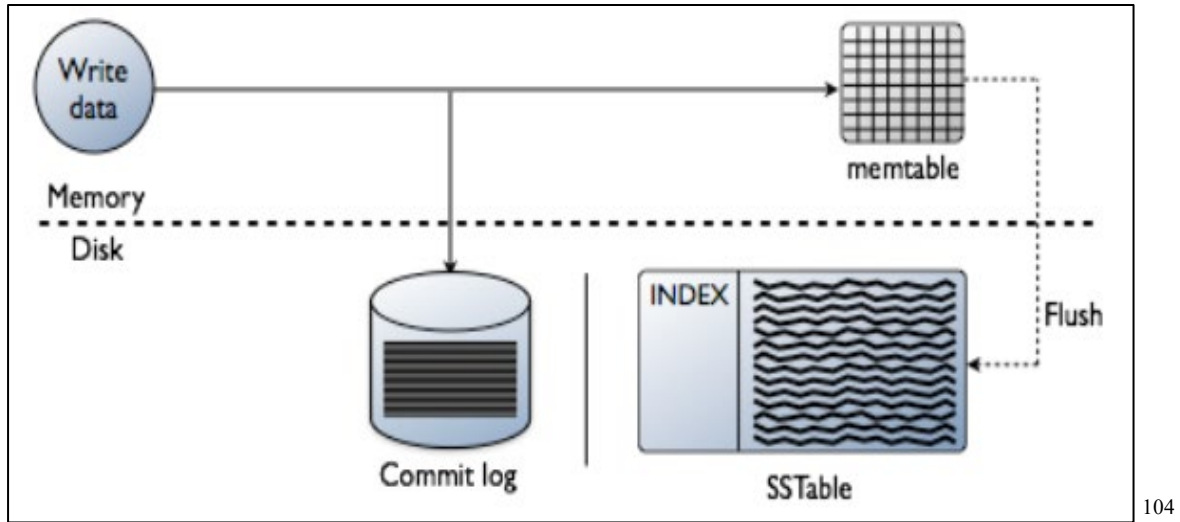
User defined compaction

a user triggers a compaction on a given set of SSTables.

103

105. Apache Cassandra performs the step of intercepting communications between a computer system (e.g., a Cassandra node) and said storage device (e.g., a disk associated with a Cassandra node).

¹⁰³ <https://cassandra.apache.org/doc/latest/cassandra/operating/compaction/>



106. Apache Cassandra performs the step of analyzing said intercepted communications. For example, the step of analyzing can entail monitoring the memtable and tracking the number of SSTables flushed to a disk and comparing it to the parameters `min_threshold` and `max_threshold` to determine whether to perform a minor compaction.

¹⁰⁴ <https://docs.datastax.com/en/cassandra-oss/3.0/cassandra/dml/dmlHowDataWritten.html>

Common options

There is a number of common options for all the compaction strategies;

enabled (default: true)

Whether minor compactions should run. Note that you can have 'enabled': true as a compaction option and then do 'nodetool enableautocompaction' to start running compactions.

tombstone_threshold (default: 0.2)

How much of the sstable should be tombstones for us to consider doing a single sstable compaction of that sstable.

tombstone_compaction_interval (default: 86400s (1 day))

Since it might not be possible to drop any tombstones when doing a single sstable compaction we need to make sure that one sstable is not constantly getting recompactd - this option states how often we should try for a given sstable.

log_all (default: false)

New detailed compaction logging, see [below <detailed-compaction-logging>](#).

unchecked_tombstone_compaction (default: false)

The single sstable compaction has quite strict checks for whether it should be started, this option disables those checks and for some usecases this might be needed. Note that this does not change anything for the actual compaction, tombstones are only dropped if it is safe to do so - it might just rewrite an sstable without being able to drop any tombstones.

only_purge_repaired_tombstone (default: false)

Option to enable the extra safety of making sure that tombstones are only dropped if the data has been repaired.

min_threshold (default: 4)

Lower limit of number of SSTables before a compaction is triggered. Not used for `LeveledCompactionStrategy`.

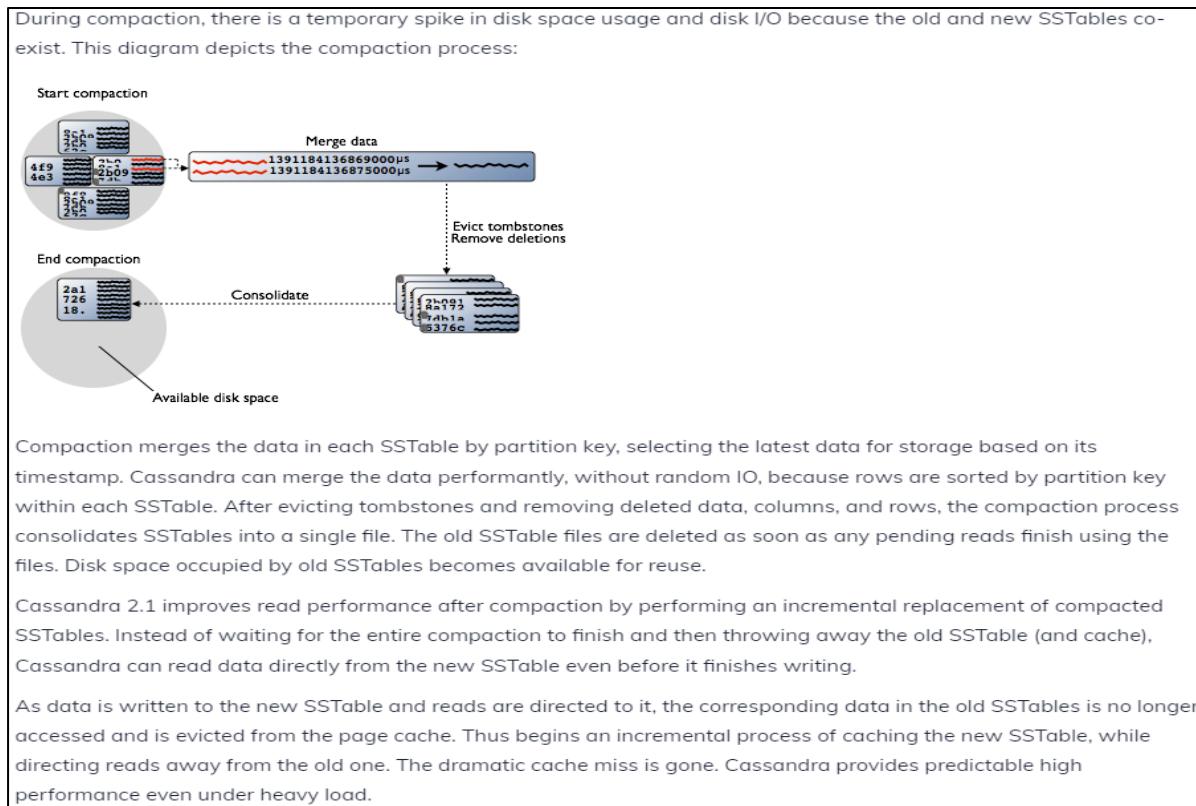
max_threshold (default: 32)

Upper limit of number of SSTables before a compaction is triggered. Not used for `LeveledCompactionStrategy`.

105

107. Apache Cassandra performs the step of reallocating at least some of said data on said storage device to enhance the performance of said storage device based on said analyzed communications. For example, the step of reallocating can occur during minor compaction and entails merging SSTables. Continuing with the example, the step of relocating enhances performance of the storage device by, e.g., reducing access (“read”) time and/or making additional disk space available for reuse.

¹⁰⁵ <https://cassandra.apache.org/doc/latest/cassandra/operating/compaction/>



106

Apache HBase

108. On information and belief, NetApp uses Apache HBase in the U.S., as set forth in Paragraph 64 and, more specifically, in at least the information associated with footnote 29.

109. Apache HBase performs the method for monitoring performance of a storage device. For example, the performance of a storage device (e.g., a disk that store HFiles and/or StoreFiles) is monitored through the use of metrics including the number of read and write requests received.

¹⁰⁶ https://docs.datastax.com/en/cassandra-oss/2.1/cassandra/dml/dml_write_path_c.html

164. HBase Monitoring

164.1. Overview

The following metrics are arguably the most important to monitor for each RegionServer for "macro monitoring", preferably with a system like [OpenTSDB](#). If your cluster is having performance issues it's likely that you'll see something unusual with this group.

HBase

- See [rs metrics](#)

107

1. An Overview of HBase I/O

The following table describes several concepts related to HBase file operations and memory (RAM) caching.

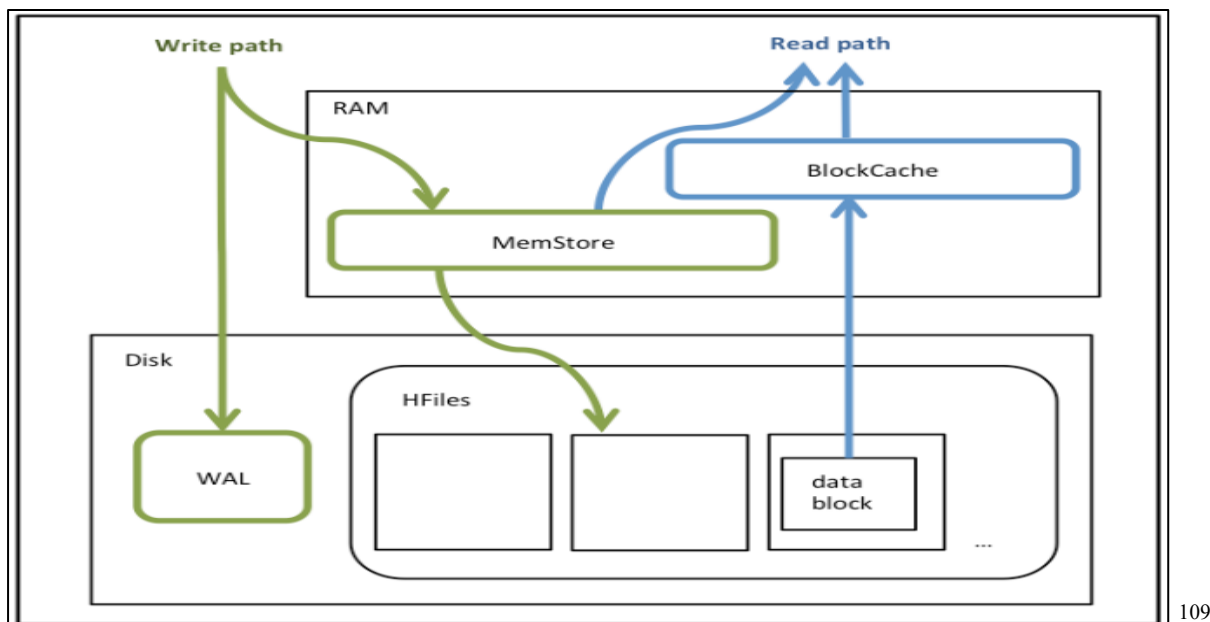
HBase Component	Description
HFile	An HFile contains table data, indexes over that data, and metadata about the data.
Block	An HBase block is the smallest unit of data that can be read from an HFile. Each HFile consists of a series of blocks. (Note: an HBase block is different than an HDFS block or other underlying file system blocks.)
BlockCache	BlockCache is the main HBase mechanism for low-latency random read operations. BlockCache is one of two memory cache structures maintained by HBase. When a block is read from HDFS, it is cached in BlockCache. Frequent access to rows in a block cause the block to be kept in cache, improving read performance.
MemStore	MemStore ("memory store") is the second of two cache structures maintained by HBase. MemStore improves write performance. It accumulates data until it is full, and then writes ("flushes") the data to a new HFile on disk. MemStore serves two purposes: it increases the total amount of data written to disk in a single operation, and it retains recently-written data in memory for subsequent low-latency reads.
Write Ahead Log (WAL)	The WAL is a log file that records all changes to data until the data is successfully written to disk (MemStore is flushed). This protects against data loss in the event of a failure before MemStore contents are written to disk.

108

¹⁰⁷ <https://hbase.apache.org/book.html#ops.monitoring>

¹⁰⁸ [https://docs.cloudera.com/HDPDocuments/HDP2/HDP-](https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.2.0/bk_Sys_Admin_Guides/content/sysadmin_HBase_io_overview.html)

[2.2.0/bk_Sys_Admin_Guides/content/sysadmin_HBase_io_overview.html](https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.2.0/bk_Sys_Admin_Guides/content/sysadmin_HBase_io_overview.html)



110. Apache HBase performs the step of intercepting communications between a computer system and said storage device. For example, a computer system is a client that functions, e.g., to send write requests, that are to be stored on a disk. When writing data, the communications between a computer system (e.g., a client) and a storage device (e.g. a disk), are intercepted by a RegionServer populating data into a MemStore prior to it being written to disk.

BlockCache and MemStore reside in random-access memory (RAM); HFiles and the Write Ahead Log are persisted to HDFS.

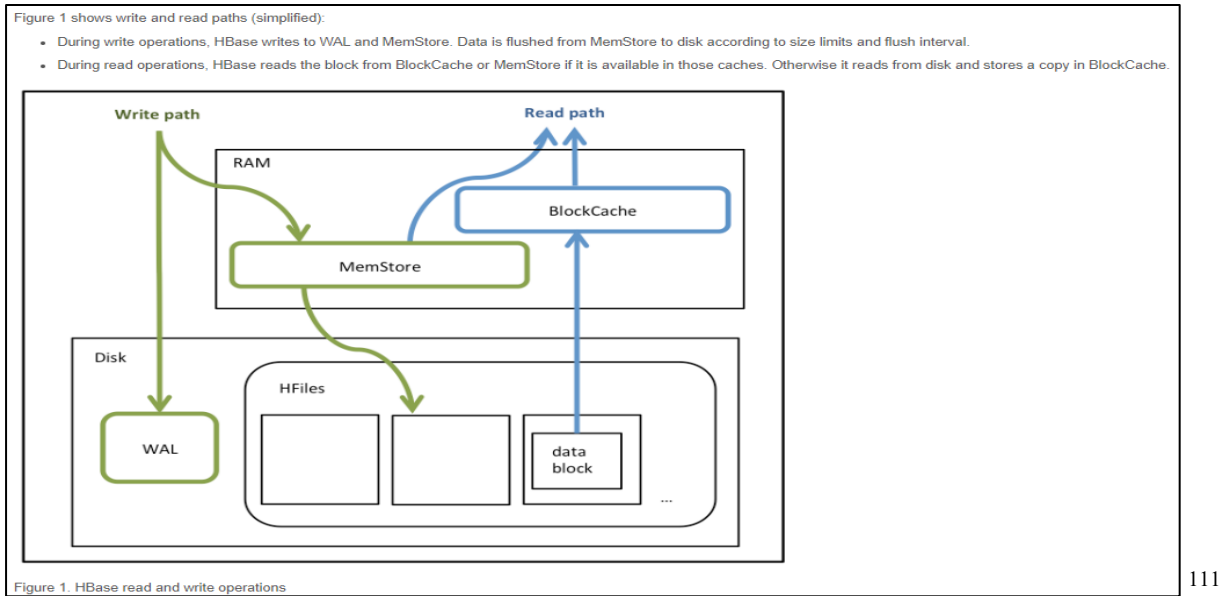
Figure 1 shows write and read paths (simplified):

- During write operations, HBase writes to WAL and MemStore. Data is flushed from MemStore to disk according to size limits and flush interval.
- During read operations, HBase reads the block from BlockCache or MemStore if it is available in those caches. Otherwise it reads from disk and stores a copy in BlockCache.

110

¹⁰⁹ https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.2.0/bk_Sys_Admin_Guides/content/sysadmin_HBase_io_overview.html

¹¹⁰ https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.2.0/bk_Sys_Admin_Guides/content/sysadmin_HBase_io_overview.html



111

111. Apache HBase performs the step of analyzing said intercepted communications. For example, Apache HBase analyzes the intercepted communications by: (1) monitoring the MemStore size; and (2) tracking the number of StoreFiles flushed to disk and comparing it to the value of the parameter `hbase.hstore.compaction.min` to determine whether to perform a compaction.

`hbase.regionserver.global.memstore.size`

Description

Maximum size of all memstores in a region server before new updates are blocked and flushes are forced. Defaults to 40% of heap (0.4). Updates are blocked and flushes are forced until size of all memstores in a region server hits `hbase.regionserver.global.memstore.size.lower.limit`. The default value in this configuration has been intentionally left empty in order to honor the old `hbase.regionserver.global.memstore.upperLimit` property if present.

Default

none

¹¹¹ https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.2.0/bk_Sys_Admin_Guides/content/sysadmin_HBase_io_overview.html

`hbase.hstore.compaction.min`

Description

The minimum number of StoreFiles which must be eligible for compaction before compaction can run. The goal of tuning `hbase.hstore.compaction.min` is to avoid ending up with too many tiny StoreFiles to compact. Setting this value to 2 would cause a minor compaction each time you have two StoreFiles in a Store, and this is probably not appropriate. If you set this value too high, all the other values will need to be adjusted accordingly. For most cases, the default value is appropriate (empty value here, results in 3 by code logic). In previous versions of HBase, the parameter `hbase.hstore.compaction.min` was named `hbase.hstore.compactionThreshold`.

Default

none

112

112. Apache HBase performs the step of reallocating at least some of said data on said storage device to enhance the performance of said storage device based on said analyzed communications. For example, compaction reallocates at least some of the data (e.g., StoreFiles) on the storage device (e.g., disk) to enhance performance of the storage device by, for example, increasing performance on read operations. The reallocating is based upon the analyzed communications including use of various parameters such as `hbase.regionserver.global.memstore.size` and `hbase.hstore.compaction.min`.

72.7.7. Compaction

Ambiguous Terminology

- A *StoreFile* is a facade of HFile. In terms of compaction, use of StoreFile seems to have prevailed in the past.
- A *Store* is the same thing as a ColumnFamily. StoreFiles are related to a Store, or ColumnFamily.
- If you want to read more about StoreFiles versus HFiles and Stores versus ColumnFamilies, see [HBASE-11316](#).

When the MemStore reaches a given size (`hbase.hregion.memstore.flush.size`), it flushes its contents to a StoreFile. The number of StoreFiles in a Store increases over time. *Compaction* is an operation which reduces the number of StoreFiles in a Store, by merging them together, in order to increase performance on read operations. Compactions can be resource-intensive to perform, and can either help or hinder performance depending on many factors.

113

113. Defendant directly infringed the '832 Patent through at least its use, promotional demonstrations, testing, repairs, and instructional guidance of Apache Kafka Streams using RocksDB Universal Style Compaction, RocksDB Universal Style Compaction, Apache Cassandra, and Apache HBase, performing at least the steps of claim 25.

¹¹² <https://hbase.apache.org/book.html>

¹¹³ <https://hbase.apache.org/2.2/book.html>

114. Defendant directly infringed the '832 Patent through at least its use, promotional demonstrations, testing, repairs, and instructional guidance of the NetApp SANtricity, as well as NetApp's systems that implement Apache Kafka Streams using RocksDB Universal Compaction, Apache Cassandra, and Apache HBase, performing at least the steps of claim 25.

115. Defendant has indirectly infringed one or more claims of the '832 Patent by knowingly and intentionally inducing others, including NetApp customers and end-users, to directly infringe, either literally or under the doctrine of equivalents, by making, using, offering to sell, selling, and/or importing into the United States products that include the infringing technology.

116. Defendant, with knowledge that at least one representative product, or the use thereof, infringes the '832 Patent at least as of May 28, 2021,¹¹⁴ knowingly and intentionally induced, direct infringement of the '832 Patent by providing the at least one representative product to end-users for use in an infringing manner. Additionally, on information and belief, Defendant has adopted a policy of not reviewing the patents of others, and instructing its officers, agents, and employees to not review the patents of others, including specifically those related to Defendant's specific industry, with the subjective belief that there was a high probability that Defendant would learn of its infringing activities, thereby remaining willfully blind to the '832 Patent at least as early as the issuance of the '832 Patent.

¹¹⁴ Plaintiffs sent Defendant correspondence on April 7, 2021, notifying them of their infringement of various patents, including some of the Asserted Patents. Thereafter, Plaintiffs and Defendant exchanged emails beginning in May 2021 regarding Defendant's infringement of various patents owned by Plaintiff. On May 28, 2021, Plaintiff sent Defendant an email with a claim chart identifying Defendant's infringement of the '832 Patent. Therefore, Defendant had knowledge of the '832 Patent at least as of May 2021.

117. Defendant has induced infringement by others, including end-users, with the intent to cause infringing acts by others or, in the alternative, with the belief that there was a high probability that others, including end-users, infringe the '832 Patent, but while remaining willfully blind to the infringement. Defendant has induced infringement by its customers and end-users by supplying them with instructions on how to operate the infringing technology in an infringing manner, while also making publicly available information on the infringing technology via Defendant's website, product literature and packaging, and other publications.¹¹⁵

118. Plaintiffs have suffered damages as a result of Defendant's direct and indirect infringement of the '832 Patent in an amount to be proven at trial.

119. Defendant has committed acts of infringement that Defendant actually knew or should have known constituted an unjustifiably high risk of infringement of at least one valid and enforceable claim of the '832 Patent. Defendant's direct and indirect infringement of the '832 Patent was willful, intentional, deliberate, and/or in conscious disregard of rights under the patent. Plaintiffs are entitled to an award of treble damages, reasonable attorney fees, and costs in bringing this action.

COUNT VI
(Infringement of the '332 Patent)

120. Paragraphs 1 through 25 are incorporated by reference as if fully set forth herein.

121. Plaintiffs have not licensed or otherwise authorized Defendant to make, use, offer for sale, sell, or import any products that embody the inventions of the '332 Patent.

¹¹⁵ See, e.g., <https://www.instaclustr.com/>; <https://www.instaclustr.com/platform/managed-apache-cassandra/>; <https://www.instaclustr.com/platform/managed-apache-kafka/>; see also supporting proof referenced herein.

122. Defendant has infringed, and continues to directly infringe, the '332 Patent, either literally or under the doctrine of equivalents, without authority and in violation of 35 U.S.C. § 271, by making, using, offering to sell, selling, and/or importing into the United States products that satisfy each and every limitation of one or more claims of the '332 Patent. Such products include NetApp systems, such as NetApp ONTAP and NetApp Trident which integrate with Kubernetes to enhance the functionality and capability of Kubernetes, NetApp Trident which was developed using Kubernetes, and/or Kubernetes for rolling updates on its own.

123. For example, Defendant has infringed, and continues to directly infringe, at least claim 1 of the '332 Patent by making, using, offering to sell, selling, and/or importing into the United States products that include NetApp systems, such as NetApp ONTAP and NetApp Trident which integrate with Kubernetes to enhance the functionality and capability of Kubernetes, NetApp Trident which was developed using Kubernetes, and/or Kubernetes for rolling updates on its own.

124. On information and belief, NetApp uses and integrates Kubernetes in its NetApp ONTAP and NetApp Trident systems. Further, on information and belief, NetApp uses Kubernetes for internal purposes.

Kubernetes integration with NetApp products

02/28/2025 | Contributors 

 Suggest changes

 PDFs

The NetApp portfolio of storage products integrates with many different aspects of a Kubernetes cluster, providing advanced data management capabilities, which enhance the functionality, capability, performance, and availability of the Kubernetes deployment.

Trident

Trident  makes it easier for enterprises to manage, protect, and move their data-rich containerized workloads running on Kubernetes within and across public clouds and on-premises. Trident provisions and provides persistent container storage using Trident from the proven and expansive NetApp storage portfolio in the public cloud and on-premises. It also offers a rich set of advanced application-aware data management functionality, such as snapshot, backup and restore, activity logs, and active cloning for data protection, disaster/data recovery, data audit, and migration use cases for Kubernetes workloads.

ONTAP

ONTAP is the NetApp multiprotocol, unified storage operating system that provides advanced data management capabilities for any application. ONTAP systems have all-flash, hybrid, or all-HDD configurations and offer many different deployment models, including engineered hardware (FAS and AFF), white-box (ONTAP Select), and cloud-only (Cloud Volumes ONTAP).

116

Trident simplifies provisioning, protecting, and moving Kubernetes apps. Free and open source, it provides automatic provisioning for NetApp ONTAP® storage as persistent volumes for Kubernetes applications.

[Watch the overview video](#) **[Read the blog](#)**

117

Why NetApp ONTAP for Kubernetes?

NetApp® ONTAP® is a versatile, flexible, robust storage operating system that delivers high availability and enterprise-class data protection functionality to critical data storage sets, regardless of the application running in the background.

118

¹¹⁶ <https://docs.netapp.com/us-en/trident/trident-get-started/k8s-netapp.html#trident>.

¹¹⁷ <https://www.netapp.com/trident/>

¹¹⁸ <https://www.netapp.com/media/80396-wp-7360.pdf>

Trident operator upgrade handling

One of the many [benefits of using the Trident operator](#) to install and upgrade Trident is the automatic handling of Trident and Kubernetes objects without disrupting existing mounted volumes. In this way, Trident can support upgrades with zero downtime, or [rolling updates](#) . In particular, the Trident operator communicates with the Kubernetes cluster to:

119



NetApp
 8 yrs

- 

Kubernetes Administrator III
 Full-time
 Jul 2022 - Present · 2 yrs 10 mos
- 

Kubernetes Administrator II
 Full-time
 Jan 2021 - Jul 2022 · 1 yr 7 mos

120

Container images and corresponding Kubernetes versions

For air-gapped installations, the following list is a reference of container images needed to install Trident. Use the `tridentctl images` command to verify the list of needed container images.

Kubernetes versions	Container image
v1.26.0, v1.27.0, v1.28.0, v1.29.0, v1.30.0, v1.31.0, v1.32.0	• docker.io/netapp/trident:25.02.0 • docker.io/netapp/trident-autosupport:25.02 • registry.k8s.io/sig-storage/csi-provisioner:v5.2.0 • registry.k8s.io/sig-storage/csi-attacher:v4.8.0 • registry.k8s.io/sig-storage/csi-resizer:v1.13.2 • registry.k8s.io/sig-storage/csi-snapshotter:v8.2.0 • registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.13.0 • docker.io/netapp/trident-operator:25.02.0 (optional)

121

¹¹⁹ <https://docs.netapp.com/us-en/trident/trident-managing-k8s/upgrade-operator-overview.html>

¹²⁰ <https://www.linkedin.com/in/chrisdavis6/>

¹²¹ https://docs.netapp.com/us-en/trident/pdfs/fullsite-sidebar/Trident_25_02_documentation.pdf

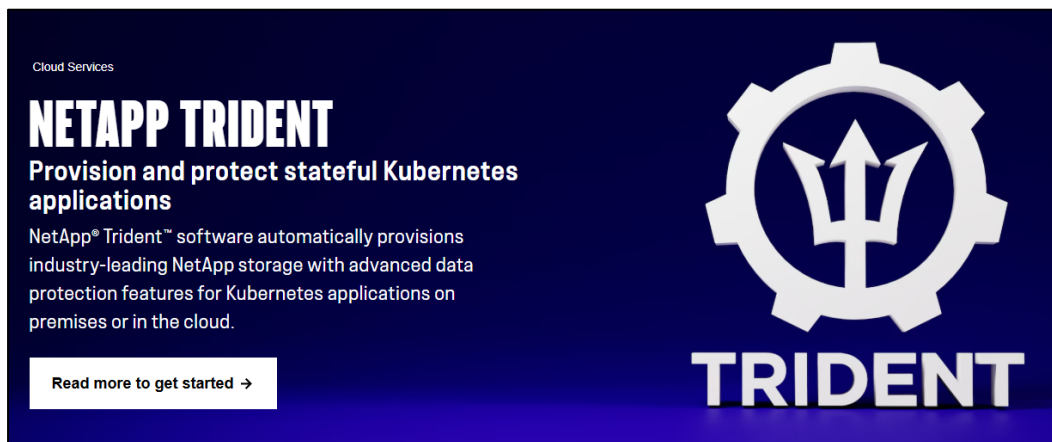
Trident

Enhancements

- **Kubernetes:**

- Added support for ONTAP ASA r2 for iSCSI.
- Added support for force detach for ONTAP-NAS volumes during Non-Graceful Node Shutdown scenarios. New ONTAP-NAS volumes will now utilize per-volume export policies managed by Trident. Provided an upgrade path for existing volumes to transition to the new export policy model on unpublish without affecting active workloads.
- Added cloneFromSnapshot annotation.
- Added support for cross namespace volume cloning.
- Enhanced iSCSI self-healing scan remediations to initiate rescans by exact host, channel, target and LUN ID.
- Added support for Kubernetes 1.32.

122



Cloud Services

NETAPP TRIDENT

Provision and protect stateful Kubernetes applications

NetApp® Trident™ software automatically provisions industry-leading NetApp storage with advanced data protection features for Kubernetes applications on premises or in the cloud.

[Read more to get started →](#)



TRIDENT

123

NETAPP TRIDENT SUPPORTS GOOGLE CLOUD NETAPP VOLUMES

July 1st, 2024

NetApp Trident, an open-source storage provisioner and orchestrator for Kubernetes used to dynamically provision and manage persistent storage for containerized applications, announces support for Google Cloud NetApp Volumes.

With Trident support, customers can easily provision, orchestrate, and attach persistent volumes (PVs) with applications on Google Kubernetes Engine, OpenShift Dedicated, and self-managed Kubernetes clusters served from Google Cloud NetApp Volumes storage pools. Built on NetApp's ONTAP technology, Google Cloud NetApp Volume provides advanced storage management capabilities like incremental snapshots with asynchronous volume replication for data protection and clones to quickly launch dev and test workloads increasing developer productivity.

124

¹²² <https://docs.netapp.com/us-en/trident/trident-rn.html>

¹²³ <https://www.netapp.com/cloud-services/trident/>

¹²⁴ <https://www.netapp.com/product-updates/trident-supports-gcnv/>

We rely on software from third parties, and a failure to properly manage our use of third-party software could result in increased costs or loss of revenue.

Many of our products are designed to include software licensed from third parties. Such third-party software includes software licensed from commercial suppliers and software licensed under public or open source licenses. We have internal processes to manage our use of such third-party software. However, if we fail to adequately manage our use of third-party software, then we may be subject to copyright infringement or other third-party claims. If we are non-compliant with a license for commercial software, then we may be required to pay penalties or undergo costly audits pursuant to the license agreement. In the case of open-source software licensed under certain "copyleft" licenses, the license itself may require, or a court-imposed remedy for non-compliant use of the open source software may require, that proprietary portions of our own software be publicly disclosed or licensed. Additionally, contract proposals, negotiations and software proposals are complex and frequently involve lengthy bidding and selection processes. We may not be able to negotiate extensions to our current third-party licenses when due for renewal or continue to secure such licenses under

commercially reasonable terms. Each of the foregoing could result in a loss of intellectual property rights, increased costs, damage to our reputation and/or a loss of revenue.

In addition, many of our products use open-source software. Such open-source software generally does not provide any warranty or contractual protection, and may be susceptible to attack from bad actors. Further, open-source software may contain vulnerabilities, which may or may not be known at the time of our inclusion of the software in a product. If a vulnerability in such software is successfully exploited, we could be subject to damages including remediation costs, reputational damage and lost revenues.

125

125. Kubernetes performs steps for hot deployment and/or redeployment in grid computing environment having one or more grid nodes, wherein the grid computing environment includes one or more grid nodes. For example, Kubernetes uses a method of rolling updates to allow for hot deployment and redeployment (rollouts and rollbacks). Using rolling updates, application containers (e.g., pods) are updated and scheduled on worker nodes of the cluster.

Kubernetes progressively rolls out changes to your application or its configuration, while monitoring application health to ensure it doesn't kill all your instances at the same time. If something goes wrong, Kubernetes will rollback the change for you. Take advantage of a growing ecosystem of deployment solutions.

126

When you deploy Kubernetes, you get a cluster. A Kubernetes cluster consists of a set of worker machines, called nodes. The worker nodes host the Pods that are the components of the application workload.

127

¹²⁵ NetApp 10-K at 26-27

¹²⁶ <https://kubernetes.io/>

¹²⁷ <https://cms-opendata-workshop.github.io/workshop2022-lesson-introcloud/01-intro/index.html>

Users expect applications to be available all the time, and developers are expected to deploy new versions of them several times a day. In Kubernetes this is done with rolling updates. A **rolling update** allows a Deployment update to take place with zero downtime. It does this by incrementally replacing the current Pods with new ones. The new Pods are scheduled on Nodes with available resources, and Kubernetes waits for those new Pods to start before removing the old Pods.

128

126. Kubernetes performs the step of adding a new version of an application release bundle in a repository server. For example, container images are software bundles shared in a registry. To rollout a new version of the software, a container image with the new version is first formed.

Once you have a [running Kubernetes cluster](#), you can deploy your containerized applications on top of it. To do so, you create a Kubernetes **Deployment**. The Deployment instructs Kubernetes how to create and update instances of your application. Once you've created a Deployment, the Kubernetes control plane schedules the application instances included in that Deployment to run on individual Nodes in the cluster.

When you create a Deployment, you'll need to specify the container image for your application and the number of replicas that you want to run. You can change that information later by updating your Deployment; [Module 5](#) and [Module 6](#) of the bootcamp discuss how you can scale and update your Deployments.

129

Images

A container image represents binary data that encapsulates an application and all its software dependencies. Container images are executable software bundles that can run standalone and that make very well defined assumptions about their runtime environment.

You typically create a container image of your application and push it to a registry before referring to it in a Pod.

¹²⁸ <https://kubernetes.io/docs/tutorials/kubernetes-basics/update/update-intro/>

¹²⁹ <https://kubernetes.io/docs/tutorials/kubernetes-basics/deploy-app/deploy-intro/>

If you don't specify a registry hostname, Kubernetes assumes that you mean the [Docker public registry](#). You can change this behaviour by setting default image registry in [container runtime](#) configuration.

130

127. Kubernetes performs the step of determining by a discovery services module which of the one or more grid nodes are running an application associated with the added new version of the application release bundle upon adding the new version of the application release bundle in the repository server. For example, a control plane has a deployment controller that ensures that the new container image is deployed all across the ReplicaSet of the cluster.

Updating an application

Users expect applications to be available all the time, and developers are expected to deploy new versions of them several times a day. In Kubernetes this is done with rolling updates. A **rolling update** allows a Deployment update to take place with zero downtime. It does this by incrementally replacing the current Pods with new ones. The new Pods are scheduled on Nodes with available resources, and Kubernetes waits for those new Pods to start before removing the old Pods.

131

Initially, the pods run the first version of your application—let's suppose its image is tagged as v1. You then develop a newer version of the app and push it to an image repository as a new image, tagged as v2. You'd next like to replace all the pods with this new version. Because you can't change an existing pod's image after the pod is created, you need to remove the old pods and replace them with new ones running the new image.

132

¹³⁰ <https://kubernetes.io/docs/concepts/containers/images/>

¹³¹ <https://kubernetes.io/docs/tutorials/kubernetes-basics/update/update-intro/>

¹³² <https://livebook.manning.com/book/kubernetes-in-action/chapter-9/13>

Deployments

A Deployment manages a set of Pods to run an application workload, usually one that doesn't maintain state.

A *Deployment* provides declarative updates for Pods and ReplicaSets.

You describe a *desired state* in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.

Each time a new Deployment is observed by the Deployment controller, a ReplicaSet is created to bring up the desired Pods. If the Deployment is updated, the existing ReplicaSet that controls Pods whose labels match `.spec.selector` but whose template does not match `.spec.template` are scaled down. Eventually, the new ReplicaSet is scaled to `.spec.replicas` and all old ReplicaSets is scaled to 0.

133

Once you have a [running Kubernetes cluster](#), you can deploy your containerized applications on top of it. To do so, you create a Kubernetes **Deployment**. The Deployment instructs Kubernetes how to create and update instances of your application. Once you've created a Deployment, the Kubernetes control plane schedules the application instances included in that Deployment to run on individual Nodes in the cluster.

Once the application instances are created, a Kubernetes Deployment controller continuously monitors those instances. If the Node hosting an instance goes down or is deleted, the Deployment controller replaces the instance with an instance on another Node in the cluster. **This provides a self-healing mechanism to address machine failure or maintenance.**

134

128. Kubernetes performs the step of notifying a client application manager associated with one or more of the determined grid nodes about adding the new version of the application release bundle along with a type of data transfer protocol to use. For example, the Deployment controller notifies Kubelet processes in worker nodes. Each Kubelet process is therefore notified of the desired configuration including the new application and associated protocol information.

¹³³ <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>

¹³⁴ <https://kubernetes.io/docs/tutorials/kubernetes-basics/deploy-app/deploy-intro/>

Deployments

A Deployment manages a set of Pods to run an application workload, usually one that doesn't maintain state.

A *Deployment* provides declarative updates for Pods and ReplicaSets.

You describe a *desired state* in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.

Each time a new Deployment is observed by the Deployment controller, a ReplicaSet is created to bring up the desired Pods. If the Deployment is updated, the existing ReplicaSet that controls Pods whose labels match `.spec.selector` but whose template does not match `.spec.template` are scaled down. Eventually, the new ReplicaSet is scaled to `.spec.replicas` and all old ReplicaSets is scaled to 0.

135

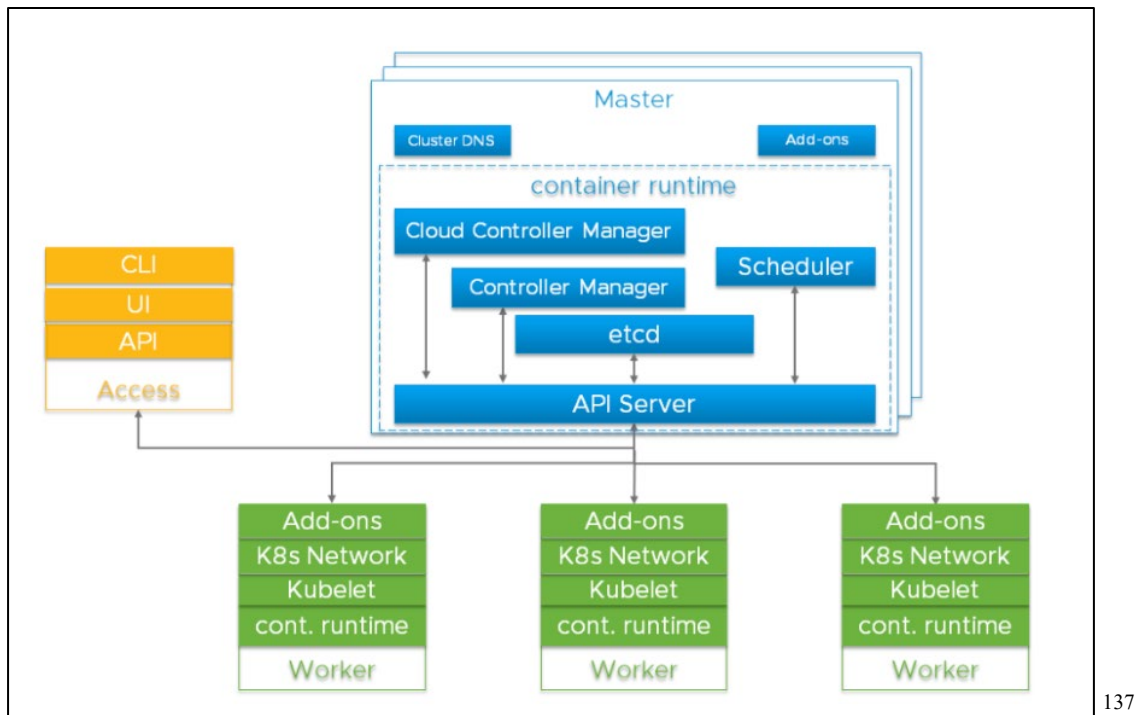
The kubelet is the primary "node agent" that runs on each node. It can register the node with the apiserver using one of: the hostname; a flag to override the hostname; or specific logic for a cloud provider.

The kubelet works in terms of a PodSpec. A PodSpec is a YAML or JSON object that describes a pod. The kubelet takes a set of PodSpecs that are provided through various mechanisms (primarily through the apiserver) and ensures that the containers described in those PodSpecs are running and healthy. The kubelet doesn't manage containers which were not created by Kubernetes.

136

¹³⁵ <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>

¹³⁶ <https://kubernetes.io/docs/concepts/overview/components/#kubelet>



137

¹³⁷ <https://kubernetes.io/blog/2018/08/03/out-of-the-clouds-onto-the-ground-how-to-make-kubernetes-production-grade-anywhere/>

```

Name:          nginx-deployment
Namespace:     default
CreationTimestamp: Thu, 30 Nov 2017 10:56:25 +0000
Labels:        app=nginx
Annotations:   deployment.kubernetes.io/revision=2
Selector:      app=nginx
Replicas:      3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:   nginx:1.16.1
      Port:    80/TCP
      Environment:  <none>
      Mounts:      <none>
      Volumes:     <none>
  Conditions:
    Type           Status  Reason
    ----           -
    Available       True    MinimumReplicasAvailable
    Progressing     True    NewReplicaSetAvailable
  OldReplicaSets:  <none>
  NewReplicaSet:   nginx-deployment-1564180365 (3/3 replicas created)
  Events:
    Type      Reason          Age    From          Message
    ----      -
    Normal    ScalingReplicaSet   2m    deployment-controller    Scaled up replica set nginx-deployment-2035384211 to 3
    Normal    ScalingReplicaSet   24s    deployment-controller    Scaled up replica set nginx-deployment-1564180365 to 1
    Normal    ScalingReplicaSet   22s    deployment-controller    Scaled down replica set nginx-deployment-2035384211 to 2
    Normal    ScalingReplicaSet   22s    deployment-controller    Scaled up replica set nginx-deployment-1564180365 to 2
    Normal    ScalingReplicaSet   19s    deployment-controller    Scaled down replica set nginx-deployment-2035384211 to 1
    Normal    ScalingReplicaSet   19s    deployment-controller    Scaled up replica set nginx-deployment-1564180365 to 3
    Normal    ScalingReplicaSet   14s    deployment-controller    Scaled down replica set nginx-deployment-2035384211 to 0

```

138

129. Kubernetes performs the step of hot deploying/redeploying the new version of the application release bundle on running one or more application servers in an associated grid node using an appropriate hot deployment plug-in based on the data transfer protocol by a respective one of the client application managers. For example, rolling updates of the new software are made with the rollout command of the kubectl interface. The image is deployed in the worker nodes according to the desired configuration by the Kubelet process.

¹³⁸ <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>

Updating an application

Rolling updates allow Deployments' update to take place with zero downtime by incrementally updating Pods instances with new ones.

Users expect applications to be available all the time, and developers are expected to deploy new versions of them several times a day. In Kubernetes this is done with rolling updates. A **rolling update** allows a Deployment update to take place with zero downtime. It does this by incrementally replacing the current Pods with new ones. The new Pods are scheduled on Nodes with available resources, and Kubernetes waits for those new Pods to start before removing the old Pods.

139

Deployments

A Deployment manages a set of Pods to run an application workload, usually one that doesn't maintain state.

A *Deployment* provides declarative updates for Pods and ReplicaSets.

You describe a *desired state* in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.

- **Declare the new state of the Pods** by updating the PodTemplateSpec of the Deployment. A new ReplicaSet is created and the Deployment manages moving the Pods from the old ReplicaSet to the new one at a controlled rate. Each new ReplicaSet updates the revision of the Deployment.

¹³⁹ <https://kubernetes.io/docs/tutorials/kubernetes-basics/update/update-intro/>.

- The `.spec.template` field contains the following sub-fields:
 - The Pods are labeled `app: nginx` using the `.metadata.labels` field.
 - The Pod template's specification, or `.spec` field, indicates that the Pods run one container, `nginx`, which runs the `nginx` [Docker Hub](#) image at version 1.14.2.
 - Create one container and name it `nginx` using the `.spec.containers[0].name` field.

140

2. To see the rollout status, run:

```
kubectl rollout status deployment/nginx-deployment
```

The output is similar to this:

```
Waiting for rollout to finish: 2 out of 3 new replicas have been updated...
```

or

```
deployment "nginx-deployment" successfully rolled out
```

Get more details on your updated Deployment:

- After the rollout succeeds, you can view the Deployment by running `kubectl get deployments`. The output is similar to this:

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
nginx-deployment	3/3	3	3	36s

¹⁴⁰ <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>.

The output is similar to this:

NAME	DESIRED	CURRENT	READY	AGE
nginx-deployment-1564180365	3	3	3	6s
nginx-deployment-2035384211	0	0	0	36s

141

```

Name:          nginx-deployment
Namespace:     default
CreationTimestamp: Thu, 30 Nov 2017 10:56:25 +0000
Labels:       app=nginx
Annotations:  deployment.kubernetes.io/revision=2
Selector:     app=nginx
Replicas:     3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:   nginx:1.16.1
      Port:    80/TCP
      Environment:  <none>
      Mounts:      <none>
      Volumes:     <none>
  Conditions:
    Type           Status  Reason
    ----           -
    Available       True    MinimumReplicasAvailable
    Progressing     True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-1564180365 (3/3 replicas created)
Events:
  Type           Reason             Age   From               Message
  ----           -
  Normal         ScalingReplicaSet   2m    deployment-controller  Scaled up replica set nginx-deployment-2035384211 to 3
  Normal         ScalingReplicaSet   24s   deployment-controller  Scaled up replica set nginx-deployment-1564180365 to 1
  Normal         ScalingReplicaSet   22s   deployment-controller  Scaled down replica set nginx-deployment-2035384211 to 2
  Normal         ScalingReplicaSet   22s   deployment-controller  Scaled up replica set nginx-deployment-1564180365 to 2
  Normal         ScalingReplicaSet   19s   deployment-controller  Scaled down replica set nginx-deployment-2035384211 to 1
  Normal         ScalingReplicaSet   19s   deployment-controller  Scaled up replica set nginx-deployment-1564180365 to 3
  Normal         ScalingReplicaSet   14s   deployment-controller  Scaled down replica set nginx-deployment-2035384211 to 0

```

142

130. Defendant directly infringes the '332 Patent through its implementation and use of Kubernetes for rolling updates, performing at least the steps of claim 1.

131. Defendant has indirectly infringed and continues to indirectly infringe one or more claims of the '332 Patent by knowingly and intentionally inducing others, including NetApp customers and end-users, to directly infringe, either literally or under the doctrine of equivalents,

¹⁴¹ <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>.

¹⁴² <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>.

by making, using, offering to sell, selling, and/or importing into the United States products that include the infringing technology.

132. Defendant, with knowledge that these products, or the use thereof, infringe the '332 Patent at least as of the date of this Complaint, knowingly and intentionally induced, and continues to knowingly and intentionally induce, direct infringement of the '332 Patent by providing these products to end-users for use in an infringing manner. Additionally, on information and belief, Defendant has adopted a policy of not reviewing the patents of others, and instructing its officers, agents, and employees to not review the patents of others, including specifically those related to Defendant's specific industry, with the subjective belief that there was a high probability that Defendant would learn of its infringing activities, thereby remaining willfully blind to the '332 Patent at least as early as the issuance of the '332 Patent.

133. Defendant has induced infringement by others, including end-users, with the intent to cause infringing acts by others or, in the alternative, with the belief that there was a high probability that others, including end-users, infringe the '332 Patent, but while remaining willfully blind to the infringement. Defendant has and continues to induce infringement by their customers and end-users by supplying them with instructions on how to operate the infringing technology in an infringing manner, while also making publicly available information on the infringing technology via Defendant's website, product literature and packaging, and other publications.¹⁴³

134. Plaintiffs have suffered damages as a result of Defendant's direct and indirect infringement of the '332 Patent in an amount to be proven at trial.

¹⁴³ See, e.g., <https://docs.netapp.com/us-en/trident/trident-get-started/k8s-netapp.html>; <https://docs.netapp.com/us-en/trident/trident-managing-k8s/upgrade-operator-overview.html>; <https://docs.netapp.com/us-en/trident/trident-rn.html#fixes-3>; <https://www.netapp.com/cloud-services/trident/>; <https://docs.netapp.com/us-en/ontap/index.html>; see also the supporting proof referenced herein.

135. Plaintiffs have suffered, and will continue to suffer, irreparable harm as a result of Defendant's infringement of the '332 Patent, for which there is no adequate remedy at law, unless Defendant's infringement is enjoined by this Court.

DEMAND FOR JURY TRIAL

Plaintiffs hereby demand a jury for all issues so triable.

PRAYER FOR RELIEF

WHEREFORE, Plaintiffs pray for relief against Defendant as follows:

- a. Entry of judgment declaring that Defendant has directly and/or indirectly infringed one or more claims of each of the Patents-in-Suit;
- b. Entry of judgment declaring that Defendant's infringement of the '416 Patent, the '571 Patent, the '686 Patent, the '488 Patent, the '332 Patent, and the '832 Patent is willful;
- c. An order pursuant to 35 U.S.C. § 283 permanently enjoining Defendant, its officers, agents, servants, employees, attorneys, and those persons in active concert or participation with them, from further acts of infringement of the '416 Patent, the '571 Patent, the '488 Patent, the '686 Patent, and the '332 Patent;
- d. An order awarding damages sufficient to compensate Plaintiffs for Defendant's infringement of the Patents-in-Suit, but in no event less than a reasonable royalty, together with interest and costs;
- e. Entry of judgment declaring that this case is exceptional and awarding Plaintiffs their costs and reasonable attorney fees under 35 U.S.C. § 285; and
- f. Such other and further relief as the Court deems just and proper.

Dated: May 9, 2025

Respectfully submitted,

/s/ Vincent J. Rubino, III

Alfred R. Fabricant

NY Bar No. 2219392

Email: ffabricant@fabricantllp.com

Peter Lambrianakos

NY Bar No. 2894392

Email: plambrianakos@fabricantllp.com

Vincent J. Rubino, III

NY Bar No. 4557435

Email: vrubino@fabricantllp.com

FABRICANT LLP

411 Theodore Fremd Avenue

Suite 206 South

Rye, New York 10580

Telephone: (212) 257-5797

Facsimile: (212) 257-5796

***ATTORNEYS FOR PLAINTIFFS
VALTRUS INNOVATIONS LTD. and
KEY PATENT INNOVATIONS LTD.***

CIVIL COVER SHEET

The JS 44 civil cover sheet and the information contained herein neither replace nor supplement the filing and service of pleadings or other papers as required by law, except as provided by local rules of court. This form, approved by the Judicial Conference of the United States in September 1974, is required for the use of the Clerk of Court for the purpose of initiating the civil docket sheet. (SEE INSTRUCTIONS ON NEXT PAGE OF THIS FORM.)

I. (a) PLAINTIFFS

Valtrus Innovations Ltd. and
Key Patent Innovations Ltd.

(b) County of Residence of First Listed Plaintiff Outside of U.S.
(EXCEPT IN U.S. PLAINTIFF CASES)

(c) Attorneys (Firm Name, Address, and Telephone Number)

Fabricant LLP, 411 Theodore Fremd Avenue, Suite
206S, Rye, NY 10580; (212) 257-5797

DEFENDANTS

NetApp, Inc.

County of Residence of First Listed Defendant _____
(IN U.S. PLAINTIFF CASES ONLY)

NOTE: IN LAND CONDEMNATION CASES, USE THE LOCATION OF
THE TRACT OF LAND INVOLVED.

Attorneys (If Known)

II. BASIS OF JURISDICTION (Place an "X" in One Box Only)

- ☐ 1 U.S. Government Plaintiff ☒ 3 Federal Question (U.S. Government Not a Party)
- ☐ 2 U.S. Government Defendant ☐ 4 Diversity (Indicate Citizenship of Parties in Item III)

III. CITIZENSHIP OF PRINCIPAL PARTIES (Place an "X" in One Box for Plaintiff and One Box for Defendant)

- | | PTF | DEF | | PTF | DEF |
|---|----------------------------|----------------------------|---|----------------------------|----------------------------|
| Citizen of This State | ☐ 1 | ☐ 1 | Incorporated or Principal Place of Business In This State | ☐ 4 | ☐ 4 |
| Citizen of Another State | ☐ 2 | ☐ 2 | Incorporated and Principal Place of Business In Another State | ☐ 5 | ☐ 5 |
| Citizen or Subject of a Foreign Country | ☐ 3 | ☐ 3 | Foreign Nation | ☐ 6 | ☐ 6 |

IV. NATURE OF SUIT (Place an "X" in One Box Only)

Click here for: [Nature of Suit Code Descriptions.](#)

CONTRACT	TORTS	FORFEITURE/PENALTY	BANKRUPTCY	OTHER STATUTES
☐ 110 Insurance ☐ 120 Marine ☐ 130 Miller Act ☐ 140 Negotiable Instrument ☐ 150 Recovery of Overpayment & Enforcement of Judgment ☐ 151 Medicare Act ☐ 152 Recovery of Defaulted Student Loans (Excludes Veterans) ☐ 153 Recovery of Overpayment of Veteran's Benefits ☐ 160 Stockholders' Suits ☐ 190 Other Contract ☐ 195 Contract Product Liability ☐ 196 Franchise	PERSONAL INJURY ☐ 310 Airplane ☐ 315 Airplane Product Liability ☐ 320 Assault, Libel & Slander ☐ 330 Federal Employers' Liability ☐ 340 Marine ☐ 345 Marine Product Liability ☐ 350 Motor Vehicle ☐ 355 Motor Vehicle Product Liability ☐ 360 Other Personal Injury ☐ 362 Personal Injury - Medical Malpractice PERSONAL INJURY ☐ 365 Personal Injury - Product Liability ☐ 367 Health Care/Pharmaceutical Personal Injury Product Liability ☐ 368 Asbestos Personal Injury Product Liability PERSONAL PROPERTY ☐ 370 Other Fraud ☐ 371 Truth in Lending ☐ 380 Other Personal Property Damage ☐ 385 Property Damage Product Liability	☐ 625 Drug Related Seizure of Property 21 USC 881 ☐ 690 Other LABOR ☐ 710 Fair Labor Standards Act ☐ 720 Labor/Management Relations ☐ 740 Railway Labor Act ☐ 751 Family and Medical Leave Act ☐ 790 Other Labor Litigation ☐ 791 Employee Retirement Income Security Act IMMIGRATION ☐ 462 Naturalization Application ☐ 465 Other Immigration Actions	☐ 422 Appeal 28 USC 158 ☐ 423 Withdrawal 28 USC 157 INTELLECTUAL PROPERTY RIGHTS ☐ 820 Copyrights ☒ 830 Patent ☐ 835 Patent - Abbreviated New Drug Application ☐ 840 Trademark ☐ 880 Defend Trade Secrets Act of 2016 SOCIAL SECURITY ☐ 861 HIA (1395ff) ☐ 862 Black Lung (923) ☐ 863 DIWC/DIWW (405(g)) ☐ 864 SSID Title XVI ☐ 865 RSI (405(g)) FEDERAL TAX SUITS ☐ 870 Taxes (U.S. Plaintiff or Defendant) ☐ 871 IRS—Third Party 26 USC 7609	☐ 375 False Claims Act ☐ 376 Qui Tam (31 USC 3729(a)) ☐ 400 State Reapportionment ☐ 410 Antitrust ☐ 430 Banks and Banking ☐ 450 Commerce ☐ 460 Deportation ☐ 470 Racketeer Influenced and Corrupt Organizations ☐ 480 Consumer Credit (15 USC 1681 or 1692) ☐ 485 Telephone Consumer Protection Act ☐ 490 Cable/Sat TV ☐ 850 Securities/Commodities/Exchange ☐ 890 Other Statutory Actions ☐ 891 Agricultural Acts ☐ 893 Environmental Matters ☐ 895 Freedom of Information Act ☐ 896 Arbitration ☐ 899 Administrative Procedure Act/Review or Appeal of Agency Decision ☐ 950 Constitutionality of State Statutes
REAL PROPERTY ☐ 210 Land Condemnation ☐ 220 Foreclosure ☐ 230 Rent Lease & Ejectment ☐ 240 Torts to Land ☐ 245 Tort Product Liability ☐ 290 All Other Real Property	CIVIL RIGHTS ☐ 440 Other Civil Rights ☐ 441 Voting ☐ 442 Employment ☐ 443 Housing/Accommodations ☐ 445 Amer. w/Disabilities - Employment ☐ 446 Amer. w/Disabilities - Other ☐ 448 Education PRISONER PETITIONS Habeas Corpus: ☐ 463 Alien Detainee ☐ 510 Motions to Vacate Sentence ☐ 530 General ☐ 535 Death Penalty Other: ☐ 540 Mandamus & Other ☐ 550 Civil Rights ☐ 555 Prison Condition ☐ 560 Civil Detainee - Conditions of Confinement			

V. ORIGIN (Place an "X" in One Box Only)

- ☒ 1 Original Proceeding ☐ 2 Removed from State Court ☐ 3 Remanded from Appellate Court ☐ 4 Reinstated or Reopened ☐ 5 Transferred from Another District (specify) ☐ 6 Multidistrict Litigation - Transfer ☐ 8 Multidistrict Litigation - Direct File

VI. CAUSE OF ACTION

Cite the U.S. Civil Statute under which you are filing (Do not cite jurisdictional statutes unless diversity):
35 U.S.C. §§ 1, et seq.
Brief description of cause:
Patent Infringement

VII. REQUESTED IN COMPLAINT:

☐ CHECK IF THIS IS A CLASS ACTION UNDER RULE 23, F.R.Cv.P. DEMAND \$ CHECK YES only if demanded in complaint:
JURY DEMAND: ☒ Yes ☐ No

VIII. RELATED CASE(S) IF ANY

(See instructions):

JUDGE Rodney Gilstrap DOCKET NUMBER 2:25-cv-80, 2:25-cv-81, 2:25-cv-516

DATE 5/9/2025 SIGNATURE OF ATTORNEY OF RECORD
/s/ Vincent J. Rubino III

FOR OFFICE USE ONLY

RECEIPT # _____ AMOUNT _____ APPLYING IFP _____ JUDGE _____ MAG. JUDGE _____
IPR2025-01490 EX2008 092