SANGFOR

Sangfor HCI

Technical Whitepaper: The Ins and Outs of The Split-brain

Product Version	N/A
Document Version	1.0
Released on	Nov. 23, 2021

Disclaimer

Products, services or features described in this document, whether wholly or in part, may be not within your purchase scope or usage scope. The products, services or features you purchase must be subject to the commercial contract and terms as agreed by you and Sangfor. Unless otherwise provided in the contract, Sangfor disclaims warranties of any kind, either express or implied, for the content of this document.

Due to product version upgrades or other reasons, the content of this document will be updated from time to time. Unless otherwise agreed, this document is used for reference only, and all statements, information, and recommendations therein do not constitute any express or implied warranties.

Technical Support

For technical support, please visit: https://www.sangfor.com/en/about-us/contact-us/technical-support

Send information about errors or any product related problem to tech.support@sangfor.com.

About This Document

This document describes the technical whitepaper of the ins and outs of the split-brain for Sangfor Hyper-Converged Infrastructure (HCI).

Intended Audience

This document is intended for:

- System / Network Administrator
- Technical User

Note Icons

English Icon	Description
⚠DANGER	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
⚠WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
⚠CAUTION	Indicates a hazardous situation, which if not avoided, could result in minor or moderate injury.
⚠NOTICE	Indicates a hazardous situation, which if not avoided, could result in settings failing to take effect, equipment damage, or data loss. NOTICE addresses practices not related to personal injury.
📖NOTE	Calls attention to important information, best practices, and tips. NOTE addresses information not related to personal injury or equipment damage.

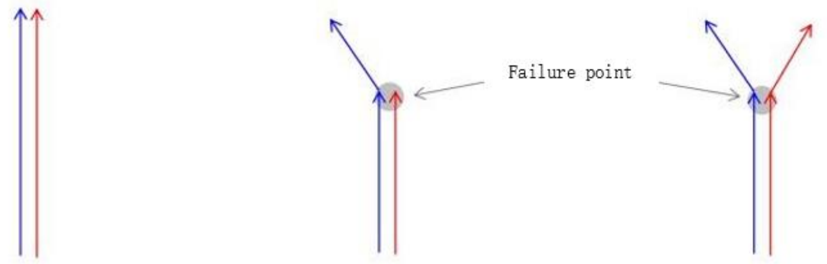
Change Log

Date	Change Description
Dec. 23, 2021	This is the first release of this document.

Contents

1 What Is Split-brain

Split-brain is a unique data inconsistency problem to distributed storage based on replicas. Split-brain is a type of inconsistent data, but inconsistent data is not necessarily a split-brain. Look at the following pictures:



Picture 1: Data is consistent

Picture 2: Data is inconsistent

Picture 3: Split-brain

Blue and red represent the data update process of the two replicas.

Picture 1: The data update of the 2 replicas is completely the same, and the 2 replicas are in a data consistent status.

Picture 2: The data update of the 2 replicas in the first phase is consistent. However, after reaching the failure point, only the blue is updated, but the red did not get the data update due to offline or other reasons, and it remains in the old status.

The 2 data replicas are inconsistent now, but these 2 replicas have not suffered a split-brain.

Picture 3: After the failure point, the 2 replicas' data is inconsistent. But the difference from picture 2 is after the failure point, the replica of blue and red have been updated separately. 2 replicas are written with different data and taken into a different path, which leads to a split-brain.

Let's look at the difference between picture 2 and picture 3. The 2 replicas in both pictures are in a data inconsistent status.

However, picture 2 is actually showing one new and one old data replica. It is not a split-brain but data inconsistent. The 2 replicas of picture 3 have been updated and formed a **Y** shape update diagram. It is a split-brain.

NOTE

Remember: **Y** shape is a sign of split-brain.

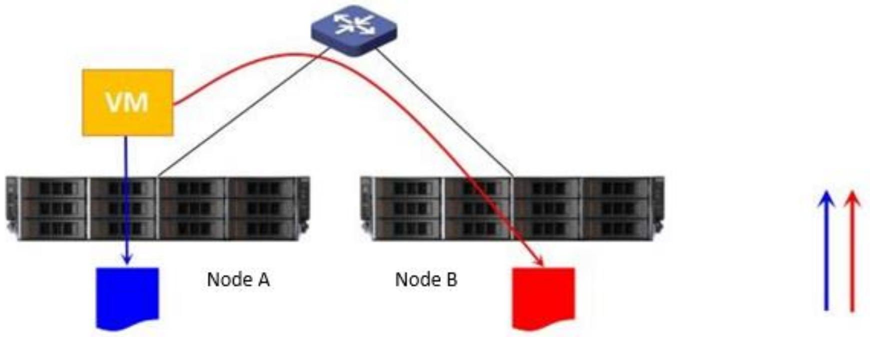
2 How Does a Split-brain Occur

How does split-brain occur? Let's look at the examples:

Suppose there are two nodes: node A and node B, and the two replicas are stored on node A and node B.

2.1 Example 1

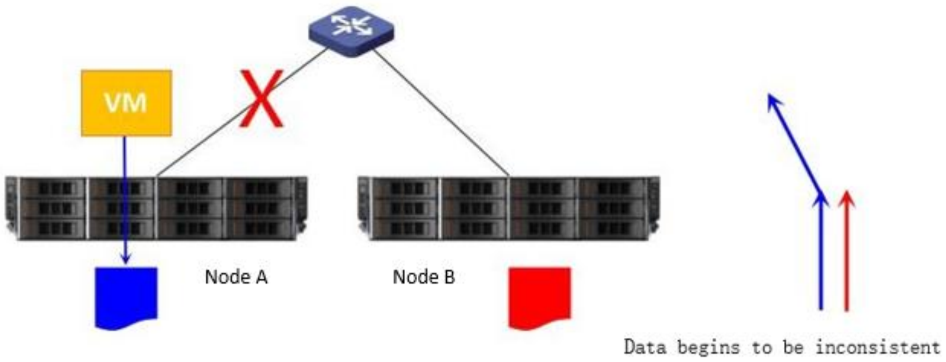
1. The virtual machine is running on node A. Everything is normal at the beginning, and the two replicas are consistent.



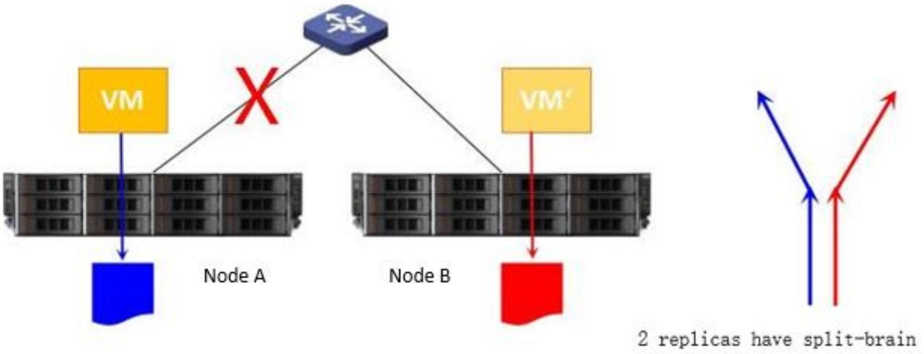
NOTE

Red and blue represent the two replicas.

2. All networks of node A are down. Node A has lost all connections with node B. At this moment, the virtual machines on node A are still running. However, because the network has down, VM can only write data to the blue replica but not the red replica. The VM has updated the blue replica, but the red replica remains in the old status. Data inconsistent happened, but there is no split-brain yet.

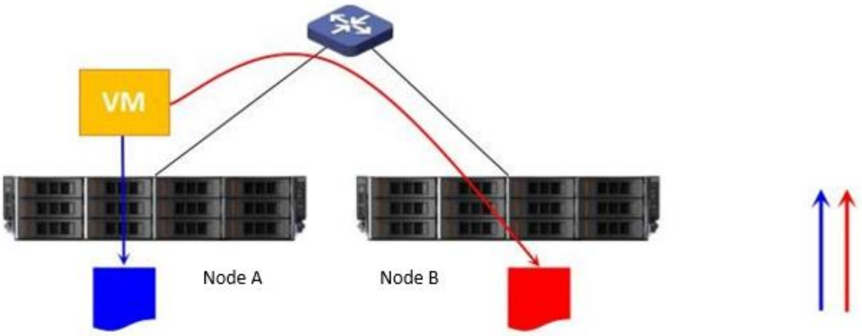


3. This virtual machine has enabled the HA mechanism, but node B didn't discover this virtual machine through detection. At this moment, the HA mechanism will be activated. Node B reactivates a new instance of the VM and marks as VM', VM' running on node B and writes new data to the red replica. Now, the red and blue replicas have been formed a **Y** shape, and there is a split-brain.



2.2 Example 2

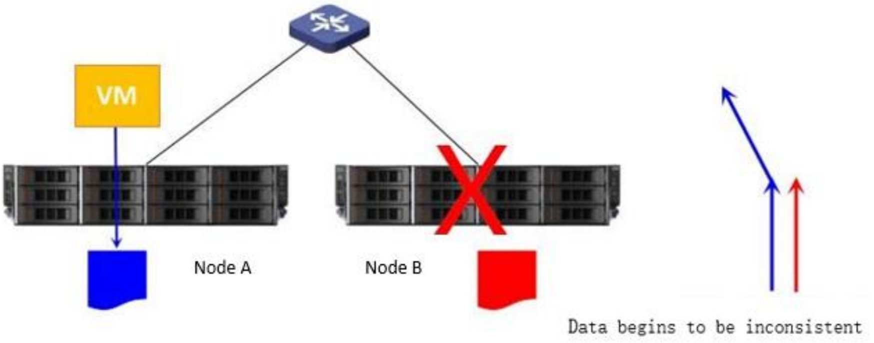
1. The virtual machine is running on node A. Everything is normal at the beginning, and the two replicas are consistent.



NOTE

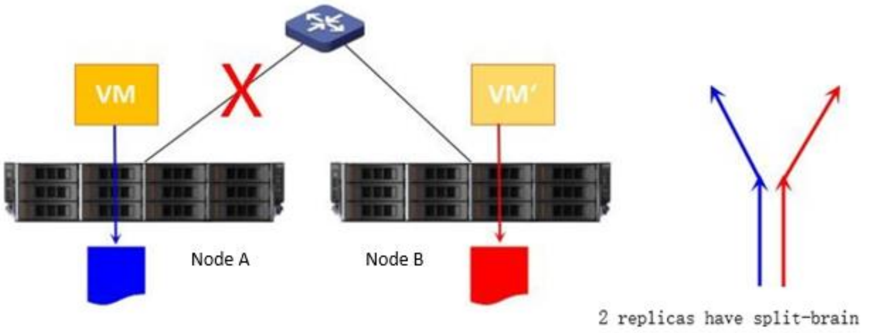
Red and blue represent the two replicas.

2. Shut down node B. The VM can only write data to the blue replica at this time. The data of the red replica cannot be updated. The two replicas are in a new status and an old status. There is data inconsistency, but no split-brain has occurred.



3. The client's server room is experiencing a power outage. The power is restored after a short while. It happened that the power cord of node A was burned out, and node B has booted up and powered on a virtual machine. Only the red replica could be written.

   At this time, two replicas appeared in the **Y** shape, and there was a split-brain.



3 How to Solve The Split-brain

If you understand the explanation of the split-brain above, the **Y** shape is the part where the data was split. Once a split-brain occurs, the split data cannot be merged. The only way is to use the data of one replica and discard the data of another replica. Therefore, once a split-brain occurs, it means data loss.

Why we cannot merge the two replicas?

Node A	Node B
Replica A	Replica A
Replica B is missing	Replica B
Replica C	Replica C
Replica D	Replica D is missing

If the two replicas are like the table above, they can be merged for sure.

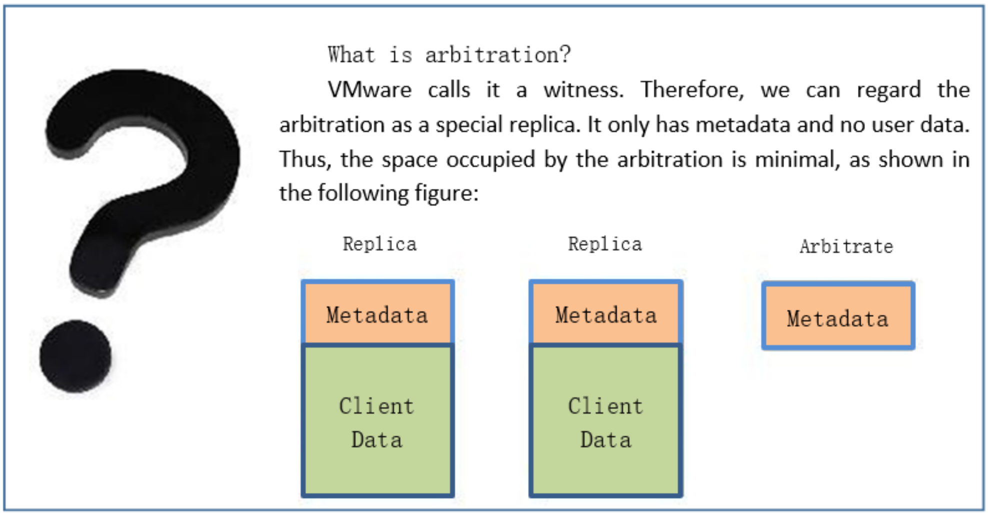
Node A	Node B
Replica A	Replica A
Replica B2	Replica B
Replica C	Replica C
Replica D	Replica D2

If the two replicas are like the table above, they cannot be merged. The replicas cannot be merged once a split-brain scenario occurs, especially for the non-txt files.

Since the occurrence of split-brain means data loss, the solution to split-brain is not to allow or prevent split-brain from occurring. Instead, the theoretical basis for preventing the occurrence of split-brain is the arbitration mechanism. The core idea of the arbitration mechanism is **The minority obeys the majority**.

For n replicas, set n-1 arbitration for them, replica + arbitration total n+(n1)=2n-1 objects. It can be written only when more than half of the objects are online.

Otherwise, it is forbidden to write. For two replicas, it means 2 replicas + 1 arbitration, and there are three objects in total. For 3 replicas, set 3 replicas + 2 arbitration = 5 objects, and so on.



What is arbitration?

VMware calls it a witness. Therefore, we can regard the arbitration as a special replica. It only has metadata and no user data. Thus, the space occupied by the arbitration is minimal, as shown in the following figure:

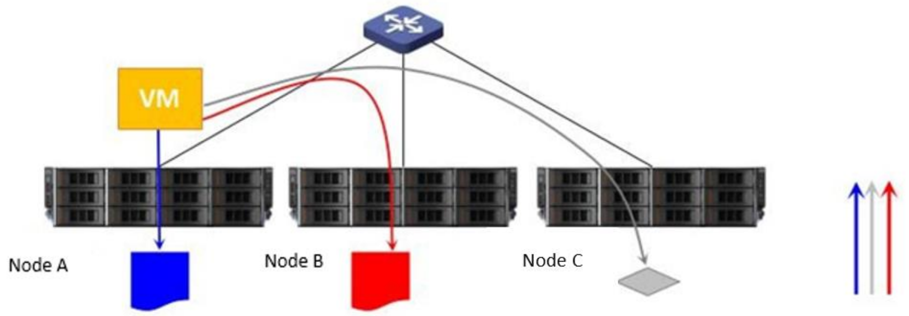
Let's take two replicas as an example to illustrate how arbitration prevents the occurrence of split-brain:

Since the 2 replicas + 1 arbitration has 3 objects, at least 3 nodes are required to support it. For example, assume that the 3 nodes are node A, node B, and node C, and the 2 replicas are located in node A and node B, and the arbitration is located in node C.

3.1 Example 1

It is the same as Example 1 in chapter 2 above.

1. The virtual machine is running on node A. Everything is normal at the beginning. 2 replicas are consistent.

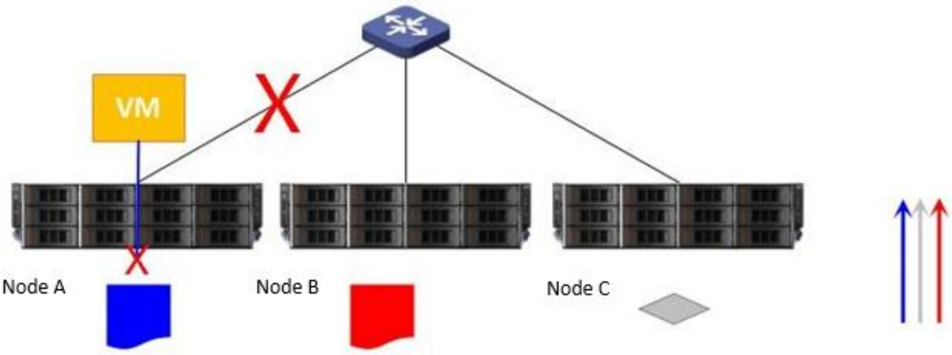


NOTE

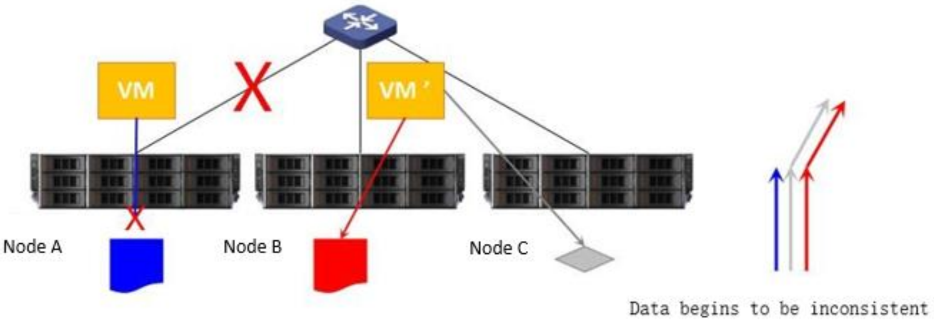
Blue and red indicate replicas, and gray diamond shape indicates arbitration.

2. All the networks of node A are disconnected, and node A has lost connection with node B and node C. At this time, the VM can only see the blue replica, and the red replica and arbitration have all lost connection with this VM. So there are 3 objects, but only 1 out of 3 objects are online, less than half.

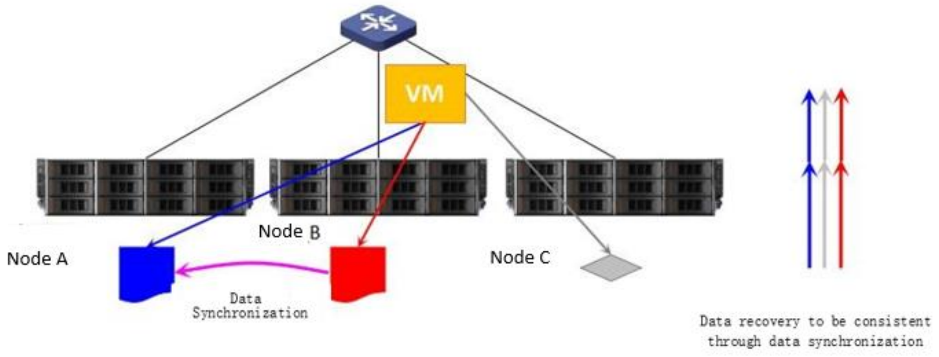
   The arbitration logic will prevent the VM from writing to the blue replica. At this time, the 3 objects are still in a consistent status.



3. Since node A is offline, nodes B and C cannot detect the virtual machine, so a virtual machine with instance VM' is powered on. Since the VM can see the red replica and arbitration simultaneously, it can write data to the red replica normally, update the metadata of the red replica, and arbitration simultaneously. Because VM' cannot write to the blue replica at this time. Therefore, the two replicas are in one new and one old status. Therefore, the **Y** shape does not appear.



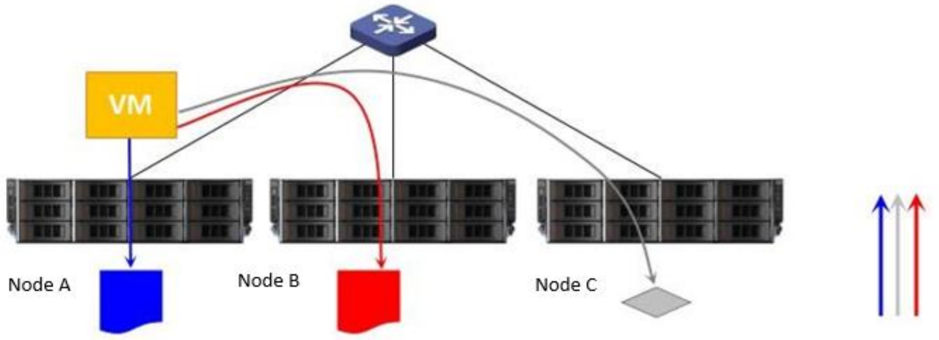
When the network of node A is restored, the suspended virtual machine instance on node A is killed, leaving the instance on node B. Since this situation is just data inconsistency, the software can automatically repair it as long as the data in the red replica is synchronized to the blue replica. Then, it can restore them to a consistent status again.



3.2 Example 2

It is the same as Example 2 in chapter 2 above.

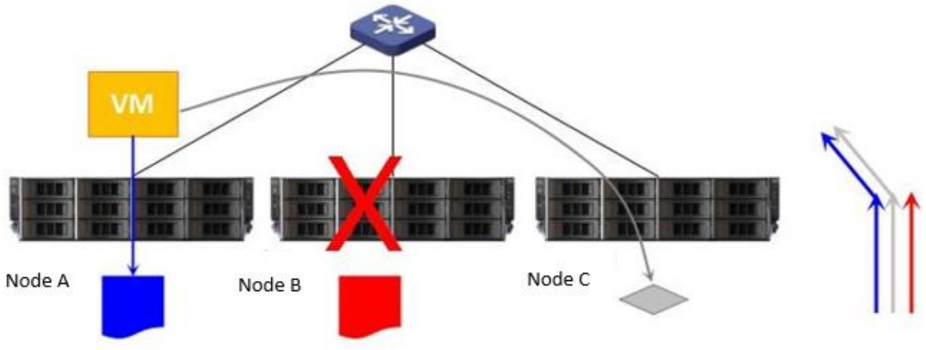
1. The virtual machine is running on node A. Everything is normal at the beginning. 2 replicas are consistent.



NOTE

Blue and red indicate replicas, and gray diamond shape indicates arbitration.

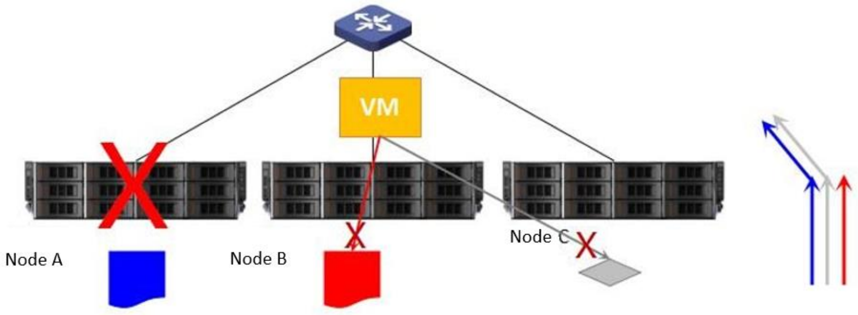
2. Shut down node B. At this time, the VM can see the blue replica and arbitration, and it can write data to the blue replica normally. The blue replica and arbitration have been updated, and the red replica is in the old status due to offline. So, it is a data inconsistency at this time.

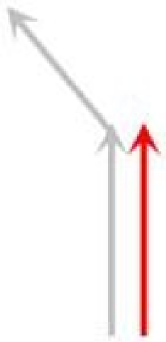


3. The client's server room is experiencing a power outage. The power is restored after a short while. Then, it happens that the power cord of node A is burned out, node B and node C are up, and node B tries to start a new virtual machine with instance VM'.

   At this time, the red replica and arbitration are both online, objects are more than half, but the arbitration will check the metadata with the red replica. Once checked, it is found that the data of the red replica is out of date, and the blue replica is the latest data, but it is offline. Therefore, the arbitration logic will prevent the writing of the red replica.

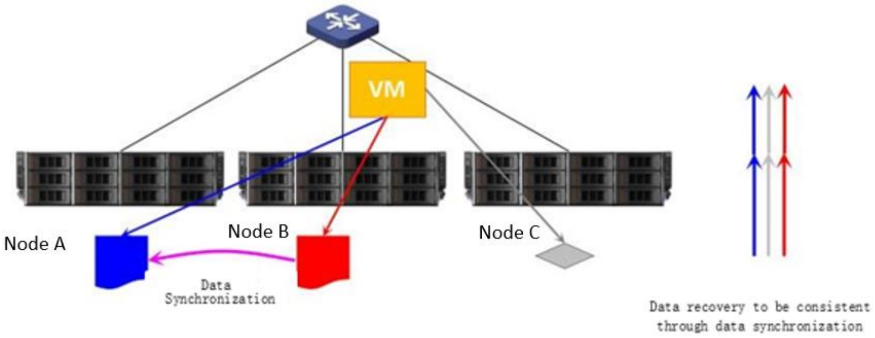
   As a result, the Y shape will not be formed, and no split-brain will occur.





The arbitration conducted a metadata comparison with the red replica and found that the red replica was old data, so writing to the red replica was prohibited, which avoided the **Y** shape and split-brain.

When the user solves the problem of node A, and it has back to online, the blue replica and arbitration are both online simultaneously. The virtual machine can resume IO (perform IO for the blue replica), and the data of the blue replica will be synchronized to the red replica to restore them to a consistent status.



From the above two examples, we can see that after adopting arbitration logic, at least more than half of the objects must be online to allow data to be written, which prevents the occurrence of split-brain.

4 Sangfor Split-brain Solutions

4.1 More than or Equal to 3 Nodes (>=3 Nodes)

An arbitration mechanism has been introduced since HCI 5.2 (VS2.3). VDI will be merged in the next version, which is VDI 5.3. There will be an arbitration mechanism when the node is >=3, and no split-brain will occur.

4.2 Two Nodes

For the two nodes environment, Sangfor HCI has implemented a different handling method and made two improvements to the reliability of the storage network:

- Each node uses two network interfaces as storage interfaces and directly connects with two network cables to form link aggregation, which significantly improves the reliability of the storage network.
- From HCI 5.3 (VS2.6) onward, it has implemented the network multiplexing module. So, when the storage network with link aggregation is disconnected, the management network or data communication network is multiplexed with the storage network. Therefore, the virtual storage will only be disconnected from the network when all the three networks (the storage network, management network, and data communication network) are disconnected.

The storage network is guaranteed to have higher reliability through the above two storage network improvement technologies, and it will significantly reduce the possibility of split-brain occurrence.

SANGFOR