Understanding Rebalancing Part 1: What Happens During Rebalancing

HP Vertica Analytic Database

Document Release Date: 8/14/2015

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Contents

About Rebalancing

After you add or remove one or more nodes in an HP Vertica cluster, the data on the existing and new nodes must be adjusted. For optimal performance, the data should be balanced across all nodes. HP Vertica calls this process *rebalancing*.

After you rebalance your cluster, the data storage and workload is balanced across all nodes in the cluster. Rebalancing is complex: CPU-, disk-, and network-intensive. Because rebalancing requires a large amount of data movement, the process can take a long time.

This is Part 1 in a two-part series about rebalancing. Part 1 explains what happens during rebalancing.

Part 2 describes the steps to take before, during, and after rebalancing to

- Prepare for rebalancing
- Monitor the rebalance operation
- Review the rebalancing results

When to Add or Remove Nodes in Your Cluster

You may need to add one or more nodes to your HP Vertica cluster when:

- The amount of data in your database has increased significantly. If you are running out of disk storage for your data, you may experience performance degradation.
- The analytic workload in your database has increased significantly. This situation may cause performance degradation.
- You need to increase the K-safety in your cluster to ensure high availability.
- You need to swap a node out of the cluster for maintenance, upgrading, or replacement. Swapping out a node does not require HP Vertica to rebalance the data across nodes.

Removing a node is less common than adding a node. You might remove a node if the cluster is over-provisioned or if you need to divert the hardware for another purpose.

Important: HP Vertica does not let you remove a node from your cluster if removing a node violates the system K-safety.

What Happens During Rebalance?

The following topics describe what takes place when you rebalance a cluster after adding or removing a node:

- Data Movement During Rebalancing
- The REFRESH Resource Pool
- Phases of Rebalancing

Data Movement During Rebalancing

Rebalancing data across a cluster is complex. HP Vertica splits segmented projections before transferring the appropriate segments to their respective destination nodes. After the rebalancing completes, on the destination nodes, the Tuple Mover merges the data segments when it next performs a mergeout.

For efficient rebalancing, existing nodes need to have free space. If the existing nodes have little free space, rebalance can still work. However, HP Vertica must perform the rebalancing in multiple stages, which can take significantly longer:

1. In the first stage, HP Vertica distributes data to the new nodes.
2. In the next stage, HP Vertica distributes data to nodes that sent data to the new nodes.
3. In subsequent stages, HP Vertica continues to send data to nodes that freed space by offloading data distributed during the previous stage.

For an example, consider a cluster with N nodes. When adding a node to a cluster, HP Vertica tries to reduce the amount of data movement requires to distribute the data evenly across all nodes. To do so, HP Vertica distributes new nodes among the existing nodes. For an illustration of this, refer to the graphic later in this section.

The amount of data that HP Vertica moves during rebalancing depends on:

- The number of nodes you have.
- The number of nodes you are adding.

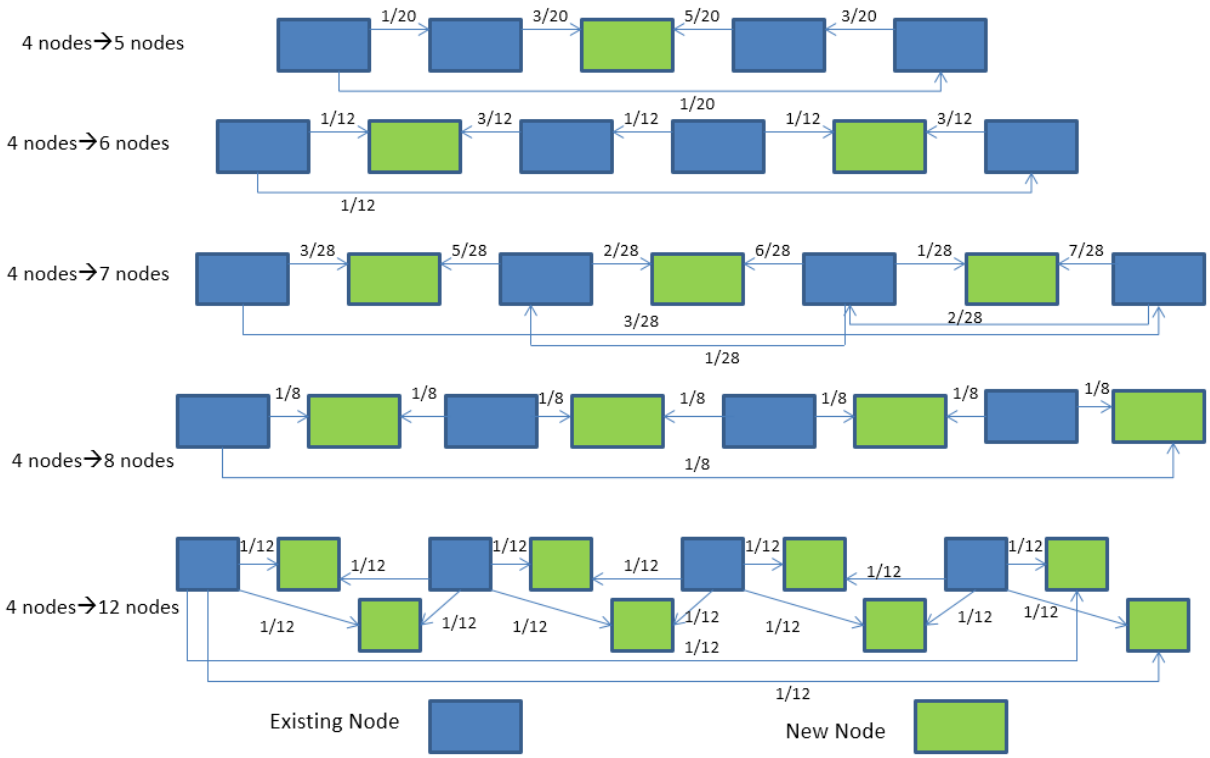
- The number of unsegmented vs. segmented projections. For example, HP Vertica copies unsegmented projections from the buddy node, because each node contains a full copy of the data.

The following graphic shows how this process works for both primary and buddy projections. The blue rectangles represent existing nodes and the green rectangles represent new nodes:

- HP Vertica inserts the nodes into the cluster at locations that minimize data movement.
- HP Vertica transfers data to both new and existing nodes. The arrows in the graphic indicate the direction of the data transfer and what percentage of moved.

  For example, the top row of the graphic shows adding one node to the four-node cluster. HP Vertica distributes new nodes in locations that minimize data movement.

  Node 2 transfers data equivalent to 3/20 of the total data in the database to the new node. Node 3 transfers data equivalent to 1/5 of the total data to the new node. In total, 13/20 of the total data moves during rebalancing.



The REFRESH Resource Pool

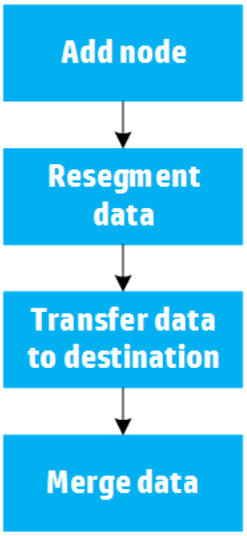
Rebalancing always runs using the built-in REFRESH resource pool. In this pool, you can specify the number of projection buddy groups that HP Vertica can rebalance at any time using the PLANNEDCONCURRENCY parameter. The MAXCONCURRENCY pool parameter has no effect on the REFRESH resource pool.

Hewlett-Packard recommends that you use the default settings for this resource pool.

Phases of Rebalancing

Because of a large amount of data movement, to save disk space, HP Vertica rebalances groups of tables and groups of projections at a time. The number of groups depends on the value of the PLANNEDCONCURRENCY configuration parameter.

The phases of rebalancing are:



The last phase, merging the data, does not happen during the rebalance. The Tuple Mover merges the data the next time it performs a mergeout.

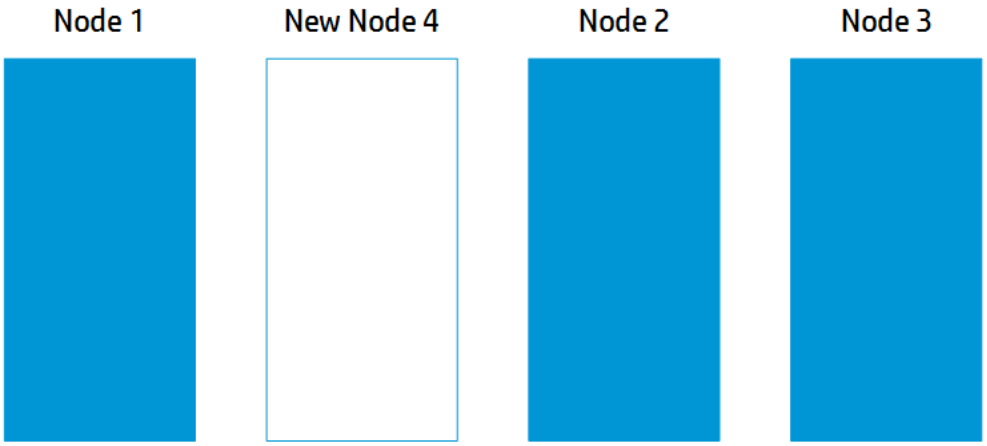
For details, see the following topics:

- Adding a Node
- Resegmenting the Data
- Transferring the Data to Destination Nodes
- Merging the Data

Adding a Node

To minimize the time it takes to move the data among the cluster nodes, HP Vertica inserts new nodes into a cluster at locations that minimize data movement. The position of the new nodes affects the performance of the rebalance. This is especially true for large clusters.

Here's how it may look when you add a one node to a three-node cluster:



Resegmenting the Data

HP Vertica reads the existing data from all nodes and looks at each table and projection.

For unsegmented projections, HP Vertica:

- Takes an X lock on each projection.
- Replicates those projections on the destination node using the following command:

```
=> CREATE PROJECTION ... UNSEGMENTED ALL NODES KSAFE
```

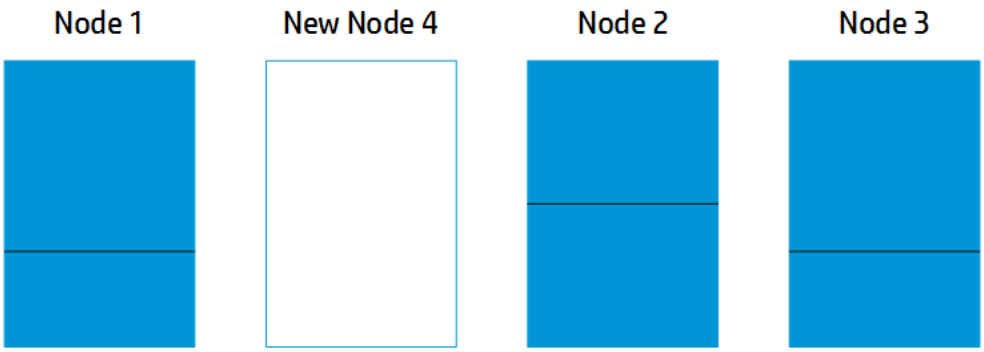
- Refreshes the projections from the buddy projections.

For segmented projections, HP Vertica:

1. Takes an S lock on the tables and an X lock on the projections.
2. Separates segments for primary, buddy, and live aggregate projections.
3. Refreshes the projections.

Segmenting the data requires a staging area, so rebalancing uses temporary storage. To use that storage efficiently, HP Vertica rebalances only a few tables and projections as a time.

After adding a node to a three-node cluster, the resegmentation might look like this:


Node 1
New Node 4
Node 2
Node 3

Transferring the Data to Destination Nodes

To balance a newly sized cluster, HP Vertica uses a hash function to determine how to distribute the data across new and existing nodes. When transferring data, HP Vertica takes an S lock and copies the unsegmented and segmented data.

Because unsegmented projections are copies of each other, the source node reads the data, and the destination node writes the data. If you have multiple new nodes, HP Vertica can transfer the unsegmented projections from multiple source nodes to multiple destination nodes in parallel. This process incurs little CPU cost.

For segmented projections, these steps are more complex. On the source nodes, HP Vertica reads, splits, and writes the segmented projections. HP Vertica requires time and disk space to perform these operations.

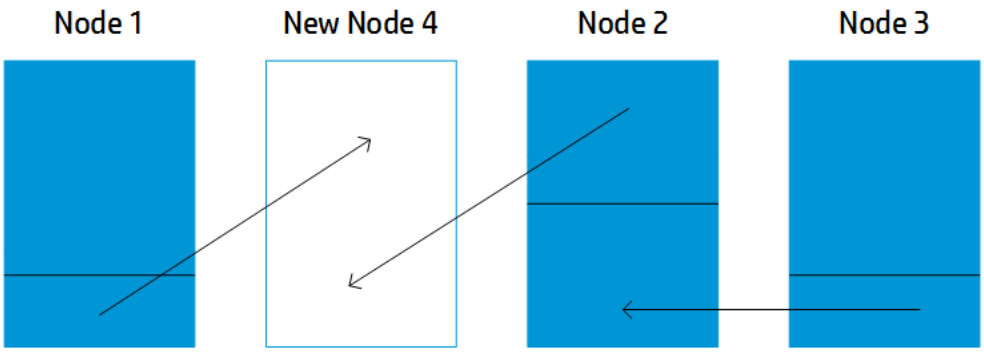
After the rebalancing completes, on the target nodes, eventually the Tuple Mover merges the data segments.

In the three-node example, you can see that the HP Vertica populates the new node with data from the adjacent nodes to minimize the amount of data transfer. After the transfer, the data is balanced across all four nodes.

In this example:

- Node 1 transfers 1/12 of the total data in the database to Node 4.
- Node 2 transfers 2/12 of the total data in the database to Node 4.
- Node 3 transfers 1/12 of the total data in the database to Node 3.

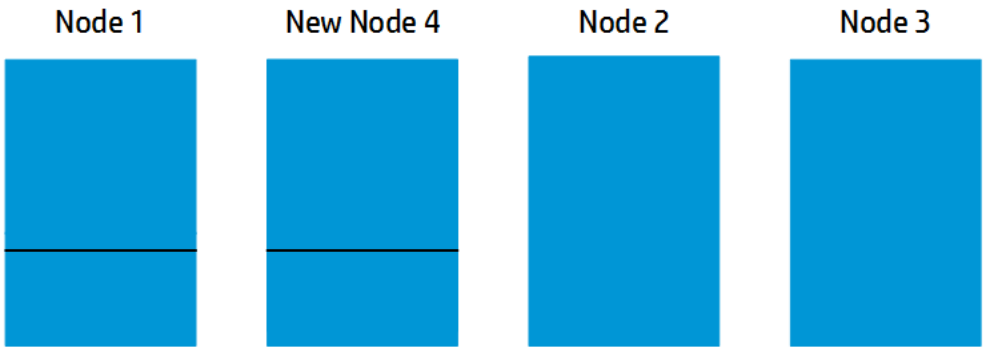
The result is that each node has 3/12 or 1/4 of the data.


Node 1
New Node 4
Node 2
Node 3

Merging the Data

After the rebalancing completes, on the destination nodes, the Tuple Mover merges the data segments during the next mergeout operation.

The following graphic illustrates this situation for a three-node cluster after adding the fourth node.



After the Tuple Mover merges the data, it refreshes all the projections. If you are removing a node, on ephemeral nodes, HP Vertica drops unneeded unsegmented projections.

How Long Does Rebalancing Take?

Rebalancing a cluster can take a long time. Any of the following factors could affect the how long it takes for rebalancing to complete:

- The number of projections.
- The amount of data and the number of rows in the projections.
- Time spent merging the data on the destination node
- Total data movement (reading and writing) of the busiest node
- Data skew
- Network throughput
- If the rebalancing process is I/O bound or network bound
- Other workloads on the cluster

Resegmenting the segmented projections and separating the ROS containers can take up to 80% of the total rebalancing time.

Before You Add or Remove a Node

Before you add or remove a node in your cluster, take these steps to optimize the performance of the rebalancing and minimize any risk of data loss:

1. Back up the database.
2. Verify that local segmentation is disabled, which is the default setting. You *must* disable local segmentation before starting a rebalance. To disable local segmentation, use the following command:

   ```
   => SELECT DISABLE_LOCAL_SEGMENTS();
   ```

3. Find out how much CPU and network bandwidth is available to run the rebalance operation. To do so, use the following HP Vertica tools:
   - `vioperf`: Measures the speed and consistency of your hard drives.
   - `vnetperf`: Measures the latency and throughput of your network between nodes.
4. Check to see if there's enough available disk space (at least 40% of the size of your database) to perform the rebalance. When moving data among the nodes, the rebalance operation uses a lot of disk space for intermediate operations.

   If there is not enough free space, HP Vertica has to perform the rebalance in multiple stages, which can take longer.

   Check the following system tables for free disk space:

- DISK_STORAGE—Amount of disk storage the database uses on each node.
- COLUMN_STORAGE—Amount of disk storage each column of each projection uses on each node.
- PROJECTION_STORAGE—Amount of disk storage each projection uses on each node.

To see the available and used disk space on your Linux file system, use the Linux `df` command:

```
$ df -h
```

To get a snapshot of each node, review the following fields in the HOST_RESOURCES system table:

```
=> SELECT host_name, disk_space_used_mb, disk_space_total_mb FROM host_resources;
```

5. To check the settings for the built-in REFRESH resource pool, enter the following statement. If necessary, adjust the settings:

```
=> SELECT name, is_internal, plannedconcurrency, maxmemorysize FROM resource_pools
   WHERE name='REFRESH';
```

6. Minimize any DML operations (COPY, INSERT, UPDATE, DELETE) on tables to be rebalanced. If rebalance has a lock on a table, the load fails. If the load has a lock on a table, the rebalancing pauses.
7. Configure the hosts you are adding to the cluster using the instructions in Configuring HP Vertica Nodes.
8. Add the hosts to the cluster using the process described in Adding Hosts to a Cluster.
9. Add the nodes to the database as described in Adding Nodes to a Database.

For More Information

For Part 2 of Understanding Rebalancing, go <insert link to Community>.

For information about rebalancing in the HP Vertica product documentation, see Rebalancing Data Across Nodes.