Mastering vRealize Operations Manager

As x86 server virtualization becomes mainstream for even the most demanding applications, the criticality of managing the health and efficiency of virtualized environments is more important than ever. vRealize Operations Manager 6.0 (vROps 6.0) is the key to simplify operations of your virtualized environment and move from being reactive to proactive.

Mastering vRealize Operations Manager 6.0 helps you streamline your processes and customize the environment to suit your needs. You will gain visibility across all devices in the network and retain full control. With easy-to-follow, step-by-step instructions and support images, you will quickly master the ability to manipulate your data and display it in a way that best suits you and the requirements of your colleagues. From the new and impressive vROps 6.0 platform architecture to troubleshooting and capacity planning, this book is aimed at ensuring you get the knowledge to manage your virtualized environment as effectively as possible.

Who this book is written for
If you are an administrator of a virtual environment and have used vROps before but want to gain a professional understanding by implementing complex tasks easily with it, then this book is for you.

What you will learn from this book
- Discover advanced vROps concepts and design your processes effectively for the underlying architecture
- Migrate from previous versions or install new versions
- Apply proven capacity management theories and techniques in practical, real-world environments
- Manipulate data and metrics to display them in the most effective way possible
- Create custom dashboards fit for any use case
- Make better use of the powerful analytics engine by bringing in external data with third-party adapters and the REST API
- Explore how policies have evolved in vROps 6.0 and how to apply them in the most effective manner

Foreword by Iwan Rahabok, VCAP-DCD, TOGAF Certified, vExpert, CTO Ambassador, VMware

Analyze and optimize your IT environment by gaining a practical understanding of vROps 6.0

In this package, you will find:

- The author’s biography
- A preview chapter from the book, Chapter 1 'vROps – Introduction, Architecture, and Availability'
- A synopsis of the book’s content
- More information on Mastering vRealize Operations Manager

About the Authors

Scott Norris is currently a senior consultant in VMware's Professional Services Organization (PSO). He specializes in multiple VMware technologies, such as ESXi, vCenter, vRA, vCD, vCOps (vROps), vRO, SRM, and vRealize Automation Application Services.

Prior to VMware, Scott worked for HP Australia and was the VMware SME for the APAC region. In this role, he did everything from breaking and fixing to architecting solutions for clients all over the region. For the last 6 years, Scott has worked exclusively on VMware products and technologies, supporting small environments, from a single server office to large federal government environments with hundreds of hosts.

Scott Norris is a 32-year-old father of two. He has been an IT professional for 11 years minus a small hiatus in which he fought and won the battle against non-Hodgkin's lymphoma. Outside of work, Scott enjoys messing around in his test lab, playing with his children, or kicking back laying the smack down in Halo on the Xbox.

This is his first book, but he runs a popular blog focused on the vRealize Suite at www.virtualiseme.net.au.

Scott can be followed on Twitter at @auScottNorris.
Christopher Slater is a managing consultant working for VMware as part of the Professional Services Organization (PSO). He specializes in SDDC technologies and methodologies such as vSphere, vRealize Operations Manager, and Infrastructure and Platform as a Service (IaaS/PaaS) through vRealize Automation. He is a VMware Certified Design Expert (VCDX-DV #102) and acquired a bachelor of IT degree from the Australian National University. Chris's primary customer base is the Australian Federal Government, which has allowed him to work in some of the largest vSphere environments in Australia.

He is in his late 20s and is grateful for the support from his wife, Nicola, and two children, Sophia and Daniel. Outside of work, Chris enjoys gaming and watching Hobbit and Frozen (for the 100th time) with his kids.

Chris is glad to join Scott in his first book writing adventure and also blogs on VMware SDDC technologies at www.definedbysoftware.com.

He can be followed on Twitter at @cslater27.
Mastering vRealize Operations Manager

When we were originally approached to write this book, the first and most obvious question that we asked was "why should we do that?" However, on reflection, we both thought about our personal experiences with the product and the real world differences it made during times when we needed to troubleshoot performance problems or report on capacity trends. We also thought about our customers for whom we had either demonstrated vROps or run knowledge transfer sessions, and how only after a few hours, the energy in the room changed as people began to grasp the power and concept of how vROps could benefit them on a daily basis.

The second reason for writing this book was because we noticed that in some environments that had the product deployed, many of the concepts, terminology, and settings were not understood well enough. As a result, customers were not getting the maximum value from their investment simply because they weren't familiar enough with the product. There is a lot of great documentation for the product, but like most other product documentation, it is generally very thin on the why aspect. For example, why should I enable some containers for capacity management and not others? Through this book and our blogs, we are making an attempt to fill this gap and to hopefully show the positive impact this product can bring to an environment.

What This Book Covers

Chapter 1, vROps - Introduction, Architecture, and Availability, begins by taking you through one of the biggest changes in vROps 6.0 from its previous versions, the new clustered platform architecture. This chapter dives straight into the new common platform design and how the controller and analytics components have been modified to work with the new GemFire cluster. We also discuss how high availability has been implemented in vROps 6.0 and how this can affect your design.

Chapter 2, Installing and Migrating vROps 6.0, walks you through a step-by-step guide for migrating (upgrading) from vC0ps 5.x to vROps 6.0. We also walkthrough the important initial vApp deployment steps and other important deployment considerations, such as designing vROps for scaling and how to add additional nodes.

Chapter 3, vRealize Operations Manager Badges, discusses in detail the critical concept of Operations Manager 6.0 badges and how they assist in effectively managing your vSphere and non-vSphere environments. We will discover the differences between badges in Operations Manager 5.x and 6.0. We dive into the details of how badges are calculated and tune them to best suit our environment.
Chapter 4, *The Merged UI*, explains in detail the major improvements in the vROps 6.0 User Interface and the merger of the old vSphere UI and Custom UI. This chapter provides a useful reference for all the major changes and provides a walkthrough for all the relevant components.

Chapter 5, *Policies in vRealize Operations Manager 6.0*, discusses the virtual topic of creating and modifying policies in vROps 6.0. We show what improvements have been made in vROps 6.0 and how policies can be used to make the smallest change to any object all the way up to an environment-wide change in a few clicks.

Chapter 6, *Capacity Management Made Easy*, dives into the detail around Operations Manager capacity management, including the major improvements made in version 6.0. We will also cover the capacity management policies in detail and understand how they need to be tuned for your environment to ensure that the recommendations are of the highest quality.

Chapter 7, *Dashboard Design*, discusses and shows what a custom dashboard is and more importantly, how to create one. This will give you the foundation to create and build custom dashboards that will suit any environment, application, or object being managed.

Chapter 8, *Reporting and Views*, covers the new vROps 6.0 features of views and reports and how they assist in easily proving any piece of information about your environment at any time. You will discover the major improvements that these features bring in to effectively manage your environment, as well as examples on how to create your own views and reports.

Chapter 9, *Super Metrics*, discusses the well-proven concept of super metrics and assists in defining the difference between metrics, objects, and attributes. After going through the various types of super metrics, we will walk through a step-by-step guide on how to create your own super metrics.

Chapter 10, *Administering vROps 6.0*, discusses the importance of how to properly administer vROps 6.0 and leverage role-based access controls (RBAC) to grant the right level of access to the right people. We will also cover how to share items such as dashboards and views for different users within your organization.

Chapter 11, *Expanding vROps with Solutions*, discusses how to get the most out of vROps 6.0 by expanding the product with solutions (previously known as adapters). We show how to install an additional solution, such as vRealize Hyperic. We will also show the interesting and useful concept of how to import your own metrics via the new REST API.
Chapter 12, Application Management in vROps 6.0, explains the power of applications and how data from different sources can be grouped together into a single application that allows simple navigation and troubleshooting.

Chapter 13, Alerting, Actions, and Recommendations, discusses the major improvements that have been made to alerting and recommendations, as well as the new concept of actions. We show how alerts, actions, and recommendations can be used to provide useful human readable alerts to the right people in real time.

Chapter 14, Just Messing Around, finishes off by showing another interesting way in which dashboards can be used to mix work and pleasure.
vRealize Operations Manager (vROps) 6.0 is a tool from VMware that helps IT administrators monitor, troubleshoot, and manage the health and capacity of their virtual environment. vROps has been developed from the stage of being a single tool to being a suite of tools known as vRealize Operations. This suite includes vCenter Infrastructure Navigator (VIN), vRealize Configuration Manager (vCM), vRealize Log Insight, and vRealize Hyperic.

Due to its popularity and the powerful analytics engine that vROps uses, many hardware vendors supply adapters (now known as solutions) that allow IT administrators to extend monitoring, troubleshooting, and capacity planning to non-vSphere systems including storage, networking, applications, and even physical devices. These solutions will be covered later in this book.

In this chapter, we will learn what's new with vROps 6.0; specifically with respect to its architecture components and platform availability.

One of the most impressive changes with vRealize Operations Manager 6.0 is the major internal architectural change of components, which has helped to produce a solution that supports both a scaled-out and high-availability deployment model. In this chapter, we will describe the new platform components and the details of the new deployment architecture. We will also cover the different roles of a vROps node (a node referring to a VM instance of vRealize Operations Manager 6.0) and simplify the design decisions needed around the complicated topics of multi-node deployment and high availability (HA).
A new, common platform design

In vRealize Operations Manager 6.0, a new platform design was required to meet some of the required goals that VMware envisaged for the product. These included:

- The ability to treat all solutions equally and to be able to offer management of performance, capacity, configuration, and compliance of both VMware and third-party solutions
- The ability to provide a single platform that can scale up to tens of thousands of objects and millions of metrics by scaling out with little reconfiguration or redesign required
- The ability to support a monitoring solution that can be highly available and to support the loss of a node without impacting the ability to store or query information

To meet these goals, **vCenter Operations Manager 5.x (vCOps)** went through a major architectural overhaul to provide a common platform that uses the same components no matter what deployment architecture is chosen. These changes are shown in the following figure:
When comparing the deployment architecture of vROps 6.0 with vCOps 5.x, you will notice that the footprint has changed dramatically. Listed in the following table are some of the major differences in the deployment of vRealize Operations Manager 6.0 compared to vRealize Operations Manager 5.x:

<table>
<thead>
<tr>
<th>Deployment considerations</th>
<th>vCenter Operations Manager 5.x</th>
<th>vRealize Operations Manager 6.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>vApp deployment</td>
<td>vApp consists of two VMs:</td>
<td>This deploys a single virtual appliance (VA), that is, the entire solution is provided in each VA.</td>
</tr>
<tr>
<td></td>
<td>• The User Interface VM</td>
<td>As many as up to 8 VAs can be deployed with this type of deployment.</td>
</tr>
<tr>
<td></td>
<td>• The Analytics VM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>There is no supported way to add additional VMs to vApp and therefore no way to scale out.</td>
<td></td>
</tr>
<tr>
<td>Scaling</td>
<td>This deployment could only be scaled up to a certain extent. If it is scaled beyond this, separate instances are needed to be deployed, which do not share the UI or data.</td>
<td>This deployment is built on the GemFire federated cluster that supports sharing of data and the UI. Data resiliency is done through GemFire partitioning.</td>
</tr>
<tr>
<td>Remote collector</td>
<td>Remote collectors are supported in vCOps 5.x, but with the installable version only. These remote collectors require a Windows or Linux base OS.</td>
<td>The same VA is used for the remote collector simply by specifying the role during the configuration.</td>
</tr>
<tr>
<td>Installable/standalone option</td>
<td>It is required that customers own MSSQL or Oracle database. No capacity planning or vSphere UI is provided with this type of deployment.</td>
<td>This deployment leverages built-in databases. It uses the same code base as used in the VA.</td>
</tr>
</tbody>
</table>

Many of these major differences will be discussed in detail in later chapters. However, the ability to support new scaled out and highly available architectures will require an administrator to consider which model is right for their environment before a vRealize Operations Manager 6.0 migration or rollout begins.
The vRealize Operations Manager component architecture
With a new common platform design comes a completely new architecture. As mentioned in the previous table, this architecture is common across all deployed nodes as well as the vApp and other installable versions. The following diagram shows the five major components of the Operations Manager architecture:

The five major components of the Operations Manager architecture depicted in the preceding figure are:

- The user interface
- Collector and the REST API
- Controller
- Analytics
- Persistence
The user interface

In vROps 6.0, the UI is broken into two components—the Product UI and the Admin UI. Unlike the vCOps 5.x vApp, the vROps 6.0 Product UI is present on all nodes with the exception of nodes that are deployed as remote collectors. Remote collectors will be discussed in more detail in the next section.

The Admin UI is a web application hosted by Pivotal tc Server (a Java application Apache web server) and is responsible for making HTTP REST calls to the Admin API for node administration tasks. The Cluster and Slice Administrator (CaSA) is responsible for cluster administrative actions such as:

- Enabling/disabling the Operations Manager cluster
- Enabling/disabling cluster nodes
- Performing software updates
- Browsing logfiles

The Admin UI is purposely designed to be separate from the Product UI and always be available for administration and troubleshooting tasks. A small database caches data from the Product UI that provides the last known state information to the Admin UI in the event that the Product UI and analytics are unavailable.

The Product UI is the main Operations Manager graphical user interface. Like the Admin UI, the Product UI is based on Pivotal tc Server and can make HTTP REST calls to the CaSA for administrative tasks. However, the primary purpose of the Product UI is to make GemFire calls to the Controller API to access data and create views, such as dashboards and reports. GemFire is part of the major underlying architectural change of vROps 6.0, which is discussed in more detail later in this chapter.
As shown in the following figure, the Product UI is simply accessed via HTTPS on TCP port 443. Apache then provides a reverse proxy to the Product UI running in Pivotal tc Server using the Apache AJP protocol.

**Collector**

The collector's role has not differed much from that in vCOps 5.x. The collector is responsible for processing data from solution adapter instances. As shown in the following figure, the collector uses adapters to collect data from various sources and then contacts the GemFire locator for connection information of one or more controller cache servers. The collector service then connects to one or more Controller API GemFire cache servers and sends the collected data.

It is important to note that although an instance of an adapter can only be run on one node at a time, this does not imply that the collected data is being sent to the controller on that node. This will be discussed in more detail later under the *Multi-node deployment and high availability* section.
Controller

The controller manages the storage and retrieval of the inventory of the objects within the system. The queries are performed by leveraging the GemFire MapReduce function that allows you to perform selective querying. This allows efficient data querying as data queries are only performed on selective nodes rather than all nodes.

We will go in detail to know how the controller interacts with the analytics and persistence stack a little later as well as its role in creating new resources, feeding data in, and extracting views.

Analytics

Analytics is at the heart of vROps as it is essentially the runtime layer for data analysis. The role of the analytics process is to track the individual states of every metric and then use various forms of correlation to determine whether there are problems.
At a high level, the analytics layer is responsible for the following tasks:

- Metric calculations
- Dynamic thresholds
- Alerts and alarms
- Metric storage and retrieval from the Persistence layer
- Root cause analysis
- **Historic Inventory Server (HIS)** version metadata calculations and relationship data

One important difference between vROps 6.0 and vCOps 5.x is that analytics tasks are now run on every node (with the exception of remote collectors). The vCOps 5.x Installable provides an option of installing separate multiple remote analytics processors for **dynamic threshold (DT)** processing. However, these remote DT processors only support dynamic threshold processing and do not include other analytics functions.

Although its primary tasks have not changed much from vCOps 5.x, the analytics component has undergone a significant upgrade under the hood to work with the new GemFire-based cache and the Controller and Persistence layers.

**Persistence**

The Persistence layer, as its name implies, is the layer where the data is persisted to a disk. The layer primarily consists of a series of databases that replace the existing vCOps 5.x **filesystem database (FSDB)** and PostgreSQL combination.

Understanding the persistence layer is an important aspect of vROps 6.0, as this layer has a strong relationship with the data and service availability of the solution. vROps 6.0 has four primary database services built on the EMC Documentum xDB (an XML database) and the original FSDB. These services include:

<table>
<thead>
<tr>
<th>Common name</th>
<th>Role</th>
<th>DB type</th>
<th>Sharded</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global xDB</td>
<td>Global data</td>
<td>Documentum xDB</td>
<td>No</td>
<td>/storage/vcops/xdb</td>
</tr>
<tr>
<td>Alarms xDB</td>
<td>Alerts and Alarms data</td>
<td>Documentum xDB</td>
<td>Yes</td>
<td>/storage/vcops/alarmxdb</td>
</tr>
<tr>
<td>Common name</td>
<td>Role</td>
<td>DB type</td>
<td>Sharded</td>
<td>Location</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------</td>
<td>---------------</td>
<td>---------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>HIS xDB</td>
<td>Historical Inventory Service data</td>
<td>Documentum xDB</td>
<td>Yes</td>
<td>/storage/vcops/hisxdb</td>
</tr>
<tr>
<td>FSDB</td>
<td>Filesystem Database</td>
<td>FSDB</td>
<td>Yes</td>
<td>/storage/db/vcops/data</td>
</tr>
<tr>
<td>CaSA DB</td>
<td>Cluster and Slice Administrator data</td>
<td>HSQLDB (HyperSQL database)</td>
<td>N/A</td>
<td>/storage/db/casa/webapp/hsqldb</td>
</tr>
</tbody>
</table>

Sharding is the term that GemFire uses to describe the process of distributing data across multiple systems to ensure that computational, storage, and network loads are evenly distributed across the cluster.

We will discuss persistence in more detail, including the concept of sharding, a little later under the Multi-node deployment and high availability section in this chapter.

**Global xDB**

Global xDB contains all of the data that, for the release of vROps, can not be sharded. The majority of this data is user configuration data that includes:

- User created dashboards and reports
- Policy settings and alert rules
- Super metric formulas (not super metric data, as this is sharded in the FSDB)
- Resource control objects (used during resource discovery)

As Global xDB is used for data that cannot be sharded, it is solely located on the master node (and master replica if high availability is enabled). More on this topic will be discussed under the Multi-node deployment and high availability section.

**Alarms xDB**

Alerts and Alarms xDB is a sharded xDB database that contains information on DT breaches. This information then gets converted into vROps alarms based on active policies.
HIS xDB
HIS xDB is a sharded xDB database that holds historical information on all resource properties and parent/child relationships. HIS is used to change data back to the analytics layer based on the incoming metric data that is then used for DT calculations and symptom/alarm generation.

FSDB
The role of the Filesystem Database is not differed much from vCOps 5.x. The FSDB contains all raw time series metrics for the discovered resources.

The FSDB metric data, HIS object, and Alarms data for a particular resource share the same GemFire shard key. This ensures that the multiple components that make up the persistence of a given resource are always located on the same node.

vRealize Operations Manager node types
vROps 6.0 contains a common node architecture. Before a node can be added to an existing cluster, a role must be selected. Although deployment will be discussed in detail in the next chapter, from a design perspective, it is important to understand the different roles and which deployment model best fits your own environment.

The master / master replica node
The master or master replica node is critical to the availability of the Operations Manager cluster. It contains all vROps 6.0 services, including UI, Controller, Analytics, Collector, and Persistence, as well as the critical services that cannot be replicated across all cluster nodes. These include:

- Global xDB
- An NTP server
- The GemFire locator

As previously discussed, Global xDB contains all of the data that we are unable to shard across the cluster. This data is critical to the successful operation of the cluster and is only located on the master node. If HA is enabled, this DB is replicated to the master replica; however, the DB is only available as read/write on the master node.
During a failure event of the master node, the master replica DB is promoted to a full read/write master. Although the process of the replica DB’s promotion can be done online, the migration of the master role during a failover does require an automated restart of the cluster. As a result, even though it is an automated process, the failure of the master node will result in a temporary outage of the Operations Manager cluster until all nodes have been restarted against the new master.

The master also has the responsibility of running both an NTP server and client. On initial configuration of the first vROps node, you are prompted to add an external NTP source for time synchronization. The master node then keeps time of this source and runs its own NTP server for all data and collector nodes to sync from. This ensures that all the nodes have the correct time and only the master/master replica requires access to the external time source.

The final component that is unique to the master role is the GemFire locator. The GemFire locator is a process that tells the starting or connecting data nodes where the currently running cluster members are located. This process also provides load balancing of queries that are passed to data nodes that then become data coordinators for that particular query. The structure of the master node is shown in the following figure:
The data node

The data node is the standard vROps 6.0 node type and is the default when adding a new node into an existing cluster. It provides the core functionality of collecting and processing data and data queries as well as extending the vROps cluster by being a member of the GemFire Federation, which in turn provides the horizontal scaling of the platform.

As shown in the following diagram, a data node is almost identical to a master/master replica node with the exception of Global xDB, the NTP server, and GemFire locator:
The remote collector node

The remote collector role is a continuation of the vCOps 5.x Installable concept, which includes using a standalone collector for remote sites or secure enclaves. Remote collectors do not process data themselves; instead, they simply forward metric data to data nodes for analytics processing.

Remote collector nodes do not run several of the core vROps components including:

- The Product UI
- Controller
- Analytics
- Persistence

As a result of not running these components, remote collectors are not members of the GemFire Federation, and although they do not add resources to the cluster, they themselves require far fewer resources to run, which is ideal in smaller remote office locations.

An important point to note is that the adapter instances will fail over other data nodes when the hosting node fails even if HA is not enabled. An exception to this is the remote collectors, as adapter instances registered to remote collectors will not automatically fail over.
Multi-node deployment and high availability

So far, we have focused on the new architecture and components of vROps 6.0 as well as mentioning the major architectural changes that the GemFire-based controller and the analytics and persistence layers have introduced. Now, before we close this chapter, we will dive down a little deeper into how data is handled in a multi-node deployment and finally, how high availability works in vROps 6.0, and which design decisions revolve around a successful deployment.

The Operations Manager's migration to GemFire

At the core of the vROps 6.0 architecture is the powerful GemFire in-memory clustering and distributed cache. GemFire provides the internal transport bus (replacing Apache ActiveMQ in vCOPs 5.x) as well as the ability to balance CPU and memory consumption across all nodes through compute pooling, memory sharing, and data partitioning. With this change, it is better to think of the Controller, Analytics, and Persistence layers as components that span nodes rather than individual components on individual nodes.
During deployment, ensure that all your vROps 6.0 nodes are configured with the same amount of vCPUs and memory. This is because from a load balancing point of view, the Operations Manager expects all nodes to have the same amount of resources as part of the controller's round-robin load balancing.

The migration to GemFire is probably the single largest underlying architectural change from vCOps 5.x, and the result of moving to a distributed in-memory database has made many of the new vROps 6.0 features possible including:

- **Elasticity and scaling**: Nodes can be added on demand, allowing vROps to scale as required. This allows a single Operations Manager instance to scale up to 8 nodes and up to 64,000 objects.
- **Reliability**: When GemFire's high availability is enabled, a backup copy of all the data is stored in both the analytics GemFire cache and the persistence layer.
- **Availability**: Even if GemFire's high availability mode is disabled, in the event of a failure, other nodes take over the failed services and the load of the failed node (assuming that the failure was not in the master node).
- **Data partitioning**: Operations Manager leverages GemFire Data partitioning to distribute data across nodes in units called buckets. A partition region will contain multiple buckets that are configured during a startup or that are migrated during a rebalance operation. Data partitioning allows the use of the GemFire MapReduce function. This function is a "data aware query," which supports parallel data querying on a subset of the nodes. The result of this is then returned to the coordinator node for final processing.

**GemFire sharding**

When we described the Persistence layer earlier, we listed the new components related to persistence in vROps 6.0 and which components were sharded and which were not. Now, it's time to discuss what sharding actually is.

GemFire sharding is the process of splitting data across multiple GemFire nodes for placement in various partitioned buckets. It is this concept in conjunction with the controller and locator service that balances the incoming resources and metrics across multiple nodes in the Operations Manager cluster. It is important to note that data is sharded per resource and not per adapter instance. For example, this allows the load balancing of incoming and outgoing data even if only one adapter instance is configured. From a design perspective, a single Operations Manager cluster could then manage a maximum configuration vCenter with up to 10,000 VMs by distributing the incoming metrics across multiple data nodes.
The Operations Manager data is sharded in both the analytics and persistence layers, which is referred to as GemFire cache sharding and GemFire persistence sharding, respectively.

Just because the data is held in the GemFire cache on one node does not necessarily result in the data shard persisting on the same node. In fact, as both layers are balanced independently, the chances of both the cache shard and persistence shard existing on the same node is $\frac{1}{N}$, where $N$ is the number of nodes.

### Adding, removing, and balancing nodes

One of the biggest advantages of the GemFire-based cluster is the elasticity of adding nodes to the cluster as the number of resources and metrics grow in your environment. This allows administrators to add or remove nodes if the size of their environment changes unexpectedly, for example, a merger with another IT department or catering for seasonal workloads that only exist for a small period of the year.

Although adding nodes to an existing cluster is something that can be done at any time, there is a slight cost incurred when doing so. As just mentioned, when adding new nodes, it is important that they are sized the same as the existing cluster nodes, which will ensure, during a rebalance operation, that the load is distributed equally between the cluster nodes.
When adding new nodes to the cluster, sometime after initial deployment, it is recommended that the Rebalance Disk option be selected under cluster management. As seen in the preceding screenshot, the warning advises that **This is a very disruptive operation that may take hours...**, and as such, it is recommended that this be a planned maintenance activity. The amount of time this operation will take will vary depending on the size of the existing cluster and the amount of data in the FSDB. As you can probably imagine, if you are adding an 8th node to an existing 7-node cluster with tens of thousands of resources, there could potentially be several TB of data that needs to be resharded over the entire cluster. It is also strongly recommended that when adding new nodes, the disk capacity and performance should match with that of the existing nodes, as the Rebalance Disk operation assumes this is the case.

This activity is not required to achieve the compute and network load balancing benefits of the new node. This can be achieved by selecting the Rebalance GemFire option that is a far less disruptive process. As per the description, this process repartitions the JVM buckets that balance the memory across all active nodes in the GemFire Federation. With the GemFire cache balanced across all the nodes, the compute and network demand should be roughly equal across all the nodes in the cluster.

Although this allows early benefit from adding a new node into an existing cluster, unless a large amount of new resources are discovered by the system shortly afterwards the majority of disk I/O is persisted, sharded data will occur on other nodes.

Apart from adding nodes, Operations Manager also allows the removal of a node at any time as long as it has been taken offline first. This can be valuable if a cluster was originally well oversized for requirement, and it considered a waste of physical computational resource. However, this task should not be taken lightly though, as the removal of a data node without enabling high availability will result in the loss of all metrics on that node. As such, it is recommended that you should generally avoid removing nodes from the cluster.

![Tip]

If the permanent removal of a data node is necessary, ensure that high availability is first enabled to prevent data loss.
High availability in vRealize Operations Manager 6.0

One of the most impressive new features that is available as part of vROps 6.0 is the ability to configure the cluster in a HA mode to prevent against data loss. Enabling HA makes two major changes to the Operations Manager cluster:

- The primary effect of HA is that all the sharded data is duplicated by the controller layer to a primary and backup copy in both the GemFire cache and GemFire persistence layers.
- The secondary effect is that the master replica is created on a chosen data node for xDB replication of Global xDB. This node is then taken over by the role of the master node in the event that the original master fails.

Operations Manager 6.0 HA design considerations

Although HA is an impressive new feature in vROps 6.0, from a design perspective, this is not a feature that should simply be enabled without proper consideration.

As mentioned earlier, both cache and persistence data is sharded per resource, not per metric or adapter. As such, when a data node is unavailable, not only can metrics not be viewed or used for analytics, but also new metrics for resources that are on effected nodes are discarded assuming that the adapter collector is operational or failed over. This fact alone would attract administrators to simply enable HA by default and it is easy to do so under vROps 6.0.

Although HA is very easy to enable, you must ensure that your cluster is sized appropriately to handle the increased load. As HA duplicates all data stored in both the GemFire cache and persistence layers, it essentially doubles the load on the system.

When designing your Operations Manager cluster, as a general rule, you will need to double the number of nodes if you are planning to enable HA. Detailed information on scaling vROps as well as the sizing calculator can be found in KB 2093783: vRealize Operations Manager Sizing Guidelines.
It is also important to consider that Operations Manager should not be deployed in a vSphere cluster where the number of vROps nodes is greater than the underlying vSphere cluster hosts. This is because there is little point enabling HA in Operations Manager if more than one node is residing on the same vSphere host at the same time.

After deploying all your vROps nodes and enabling HA, ensure that a DRS affinity rule is created to keep all nodes on separate vSphere hosts under normal operation. This can be achieved with a DRS "separate virtual machine" or a "Virtual Machines to Hosts" affinity rule.

How does HA and data duplication work?

As we just said, HA duplicates all incoming resource data so that two copies exist instead of one in both the GemFire cache and persistence layer. This is done by creating a secondary copy of each piece of data that is used in queries if the node hosting a primary copy is unavailable.

It is important to note that HA is simply creating a secondary copy of each piece of data, and as such, only one node failure can be sustained at a time \((N-1)\) without data loss regardless of the cluster size. If a node is down, a new secondary shard of the data is not created unless the original node is removed from the cluster permanently.

When a failed node becomes available again, a node is placed into the recovery mode. During this time, data is synchronized with the other cluster members and when the synchronization is complete, the node is returned to the active status.
Let's run through this process using the preceding figure for an example of how the incoming data or the creation of a new object is handled in an HA configuration. In the preceding figure, R3 represents our new resource and R3’ represents the secondary copy:

1. A running adapter instance receives data from vCenter as it is required to create a new resource for the new object, and a discovery task is created.
2. The discovery task is passed to the cluster. This task could be passed to any one node in the cluster and once it is assigned, that node is responsible for completing the task.
3. A new analytics item is created for the new object in the GemFire cache on any node in the cluster.
4. A secondary copy of the data is created on a different node to protect against failure.
5. The system then saves the data to the persistence layer. The object is created in the inventory (HIS) and its statistics are stored in the FSDB.
6. A secondary copy of the saved (GemFire persistence sharding) HIS and FSDB data is stored on a different node to protect against data loss.

**Summary**

In this chapter, we discussed the new common platform architecture design and how Operations Manager 6.0 differs from Operations Manager 5.x. We also covered the major components that make up the Operations Manager 6.0 platform and the functions that each of the component layers provide. We then moved on to the various roles of each node type, and finally, how multi-node and HA deployment functions and what design considerations are needed to be taken into account when designing these environments. In the next chapter, we will cover how to deploy vROps 6.0 based on this new architecture.
Where to buy this book

You can buy Mastering vRealize Operations Manager from the Packt Publishing website. Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.