



Redundancy

Cisco CallManager (Release 3.0 and later) provides several forms of redundancy:

- Database redundancy—The Cisco CallManagers in a cluster maintain backup copies of their shared database.
- Call-processing redundancy—Using Cisco CallManager groups, you can designate backup Cisco CallManagers to handle call processing for a disabled Cisco CallManager in a form of redundancy known as device failover.
- Media resource redundancy
- CTI redundancy

This section covers the following topics:

- [Cisco CallManager Redundancy Groups, page 6-2](#)
- [Database Redundancy, page 6-6](#)
- [Media Resource Redundancy, page 6-6](#)
- [CTI Redundancy, page 6-7](#)
- [Where to Find More Information, page 6-7](#)

Cisco CallManager Redundancy Groups

Groups and clusters form logical collections of Cisco CallManagers and their associated devices. Groups and clusters do not necessarily relate to the physical locations of any of their members.

A cluster is a set of Cisco CallManagers that share a common database. When you install and configure the Cisco CallManager software, you specify which servers and which Cisco CallManagers belong to the same cluster, and you specify which server houses the publisher database.

A group is a prioritized list of up to three Cisco CallManagers. You can associate each group with one or more device pools to provide call processing redundancy. You use Cisco CallManager Administration to define the groups, to specify which Cisco CallManagers belong to each group, and to assign a Cisco CallManager group to each device pool.

Cisco CallManager Groups

A Cisco CallManager group comprises a prioritized list of up to three Cisco CallManagers. Each group must contain a primary Cisco CallManager, and it may contain one or two backup Cisco CallManagers. The order in which you list the Cisco CallManagers in a group determines the priority order.

Cisco CallManager groups provide both redundancy and recovery:

- *Failover*—Occurs when the primary Cisco CallManager in a group fails, and the devices reregister with the backup Cisco CallManager in that group.
- *Failback*—Occurs when a failed primary Cisco CallManager comes back into service, and the devices in that group reregister with the primary.

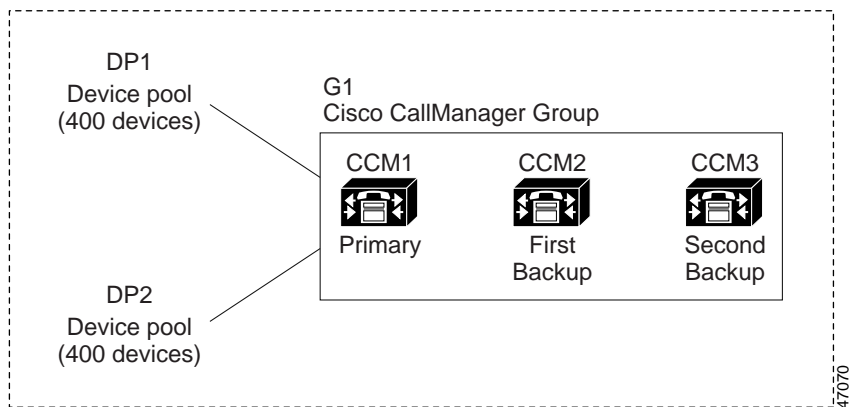
Under normal operation, the primary Cisco CallManager in a group controls call processing for all the registered devices (such as phones and gateways) associated with that group.

If the primary Cisco CallManager fails for any reason, the first backup Cisco CallManager in the group takes control of the devices that were registered with the primary Cisco CallManager. If you specify a second backup Cisco CallManager for the group, it takes control of the devices if both the primary and the first backup Cisco CallManagers fail.

When a failed primary Cisco CallManager comes back into service, it takes control of the group again, and the devices in that group automatically reregister with the primary Cisco CallManager.

You associate devices with a Cisco CallManager group by using device pools. You can assign each device to one device pool and associate each device pool with one Cisco CallManager group. You can combine the groups and device pools in various ways to achieve the desired level of redundancy. For example, [Figure 6-1](#) shows a simple system with three Cisco CallManagers in a single group controlling 800 devices.

Figure 6-1 Cisco CallManager Group



[Figure 6-1](#) depicts Cisco CallManager group G1 assigned with two device pools, DP1 and DP2. CCM1, as the primary Cisco CallManager in group G1, controls all 800 devices in DP1 and DP2 under normal operation. If CCM1 fails, control of all 800 devices transfers to CCM2. If CCM2 also fails, control of all 800 devices transfers to CCM3.

The configuration in [Figure 6-1](#) provides call processing redundancy, but it does not distribute the call processing load very well among the three Cisco CallManagers in the example. For information on load balancing, see the [“Distributing Devices for Redundancy and Load Balancing”](#) section on page 6-4.

Distributing Devices for Redundancy and Load Balancing

Cisco CallManager groups provide both call processing redundancy and distributed call processing. How you distribute devices, device pools, and Cisco CallManagers among the groups determines the level of redundancy and load balancing in your system.

In most cases, you would want to distribute the devices in a way that prevents the other Cisco CallManagers from becoming overloaded if one Cisco CallManager in the group fails. [Figure 6-2](#) shows one possible way to configure the Cisco CallManager groups and device pools to achieve both distributed call processing and redundancy for a system of three Cisco CallManagers and 800 devices.

Figure 6-2 Redundancy Combined with Distributed Call Processing

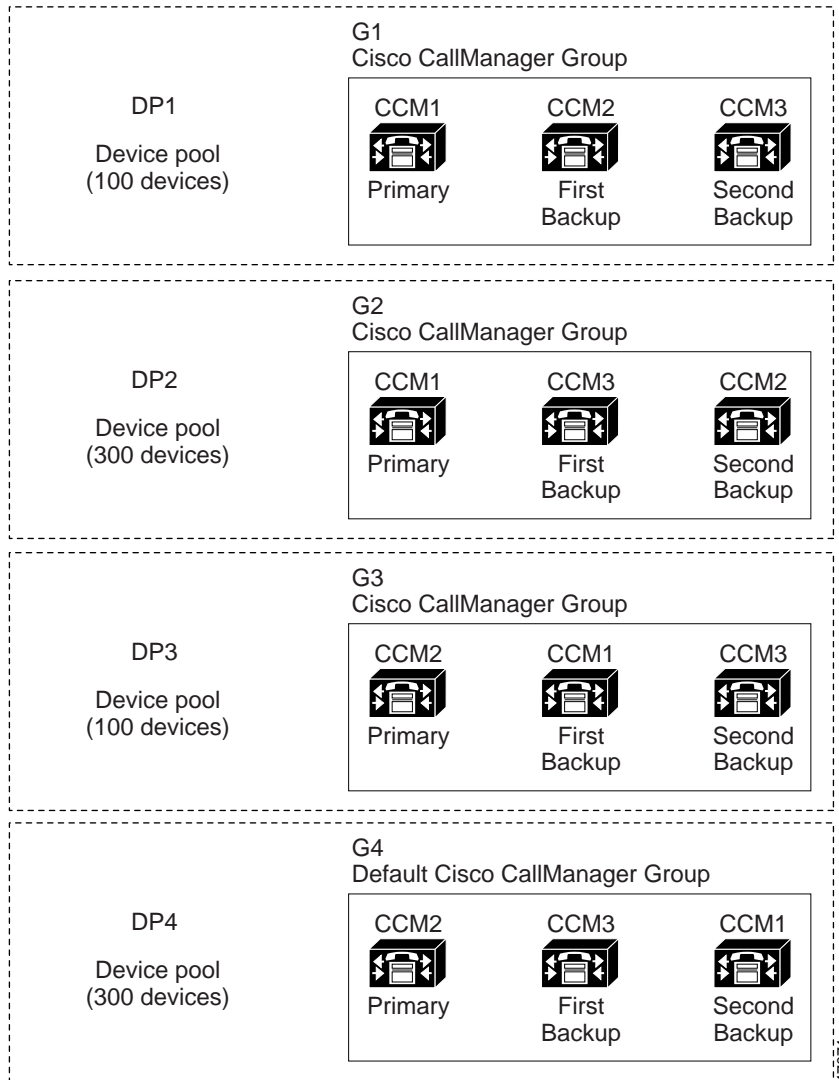


Figure 6-2 depicts the Cisco CallManager groups configured and assigned to device pools, so that Cisco CallManager CCM1 is the primary controller in two groups, G1 and G2. If CCM1 fails, the 100 devices in device pool DP1 reregister

with CCM2, and the 300 devices in DP2 reregister with CCM3. Similarly, CCM2 serves as the primary controller of groups G3 and G4. If CCM2 fails, the 100 devices in DP3 reregister with CCM1, and the 300 devices in DP4 reregister with CCM3. If CCM1 and CCM2 both fail, all devices reregister with CCM3.

For more information on distributed call processing, see the [“Balanced Call Processing” section on page 5-5](#).

Database Redundancy

When you make configuration changes in Cisco CallManager Administration, the publisher server initially stores those changes in its local database. The publisher then sends the new data to all the subscriber servers in the cluster, so that they can update their local copies of the database. This mechanism ensures consistency of the configuration database across all servers in the cluster. It also provides database redundancy because the subscriber servers can continue to operate from their read-only local copies of the database even if the publisher becomes unavailable for any reason.

Database redundancy also provides for the propagation and replication of run-time data such as registration of IP phones, gateways, and digital signal processor (DSP) resources. All servers in the cluster share this run-time data, thus ensuring optimum routing of calls between members of the cluster and associated gateways.

Media Resource Redundancy

Media resource lists provide media resource redundancy by specifying a prioritized list of media resource groups. An application can select required media resources from among the available ones according to the priority order defined in the media resource list. For more information on media resource redundancy, see the [“Media Resource Management” section on page 16-1](#).

CTI Redundancy

Computer telephony integration (CTI) provides an interface between computer-based applications and telephony functions. CTI uses various redundancy mechanisms to provide recovery from failures in any of the following major components:

- Cisco CallManager
- CTI Manager
- Applications that use CTI

CTI uses Cisco CallManager redundancy groups to provide recovery from Cisco CallManager failures. To handle recovery from failures in CTI Manager itself, CTI allows you to specify primary and backup CTI Managers for the applications that use CTI. Finally, if an application fails, the CTI Manager can redirect calls intended for that application to a forwarding directory number.

Where to Find More Information

Related Topics

- [Clustering, page 5-1](#)
- [Media Resource Management, page 16-1](#)

Additional Cisco Documentation

- *Cisco IP Telephony Network Design Guide*

