Digital Messaging Server Guide

DMS Scalability

12/19/2025

DMS Scalability

Digital Messaging Server allows a scalable functional mode that can be used when a flow of inbound messages from a media channel exceeds the processing capacity of a standalone DMS. The scalable functional mode is implemented by running multiple DMS instances. The DMS instances that participate in the scalable cluster of DMS's share the load of processing messages between them.

Prerequisites

- The DMS Scalability feature is supported in the following versions
  - 9.1.007.06
  - 9.1.008.02
  - 9.1.008.06
  - 9.1.008.07
- Genesys Driver for use with Genesys Hub 9.1.007.06 or higher

Implementation

The scalable functional mode uses the Redis server as a central data exchange element for the DMS instances of the DMS cluster. Redis server version 6.2 and Redisson (Redis client) are used in the implementation.

Two types of DMS instances participate in the processing of messages:

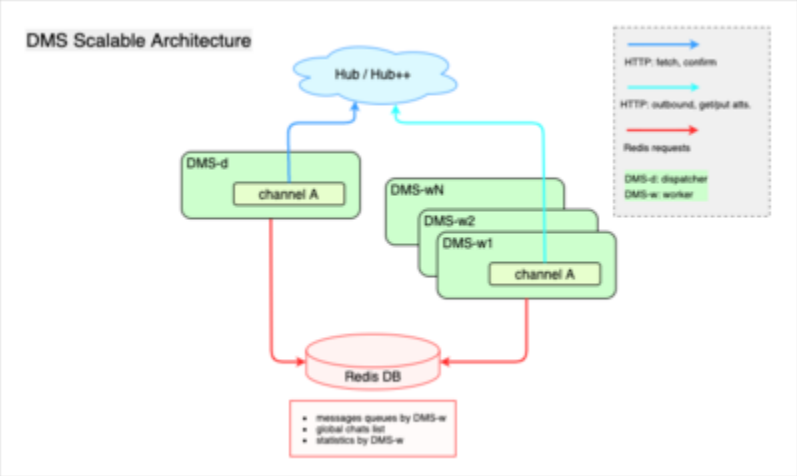
- Dispatcher (DMS-d)
- Workers (DMS-w1, DMS-w2, et al.)

All DMS instances involved in one scalable cluster are restricted to handle one media channel.

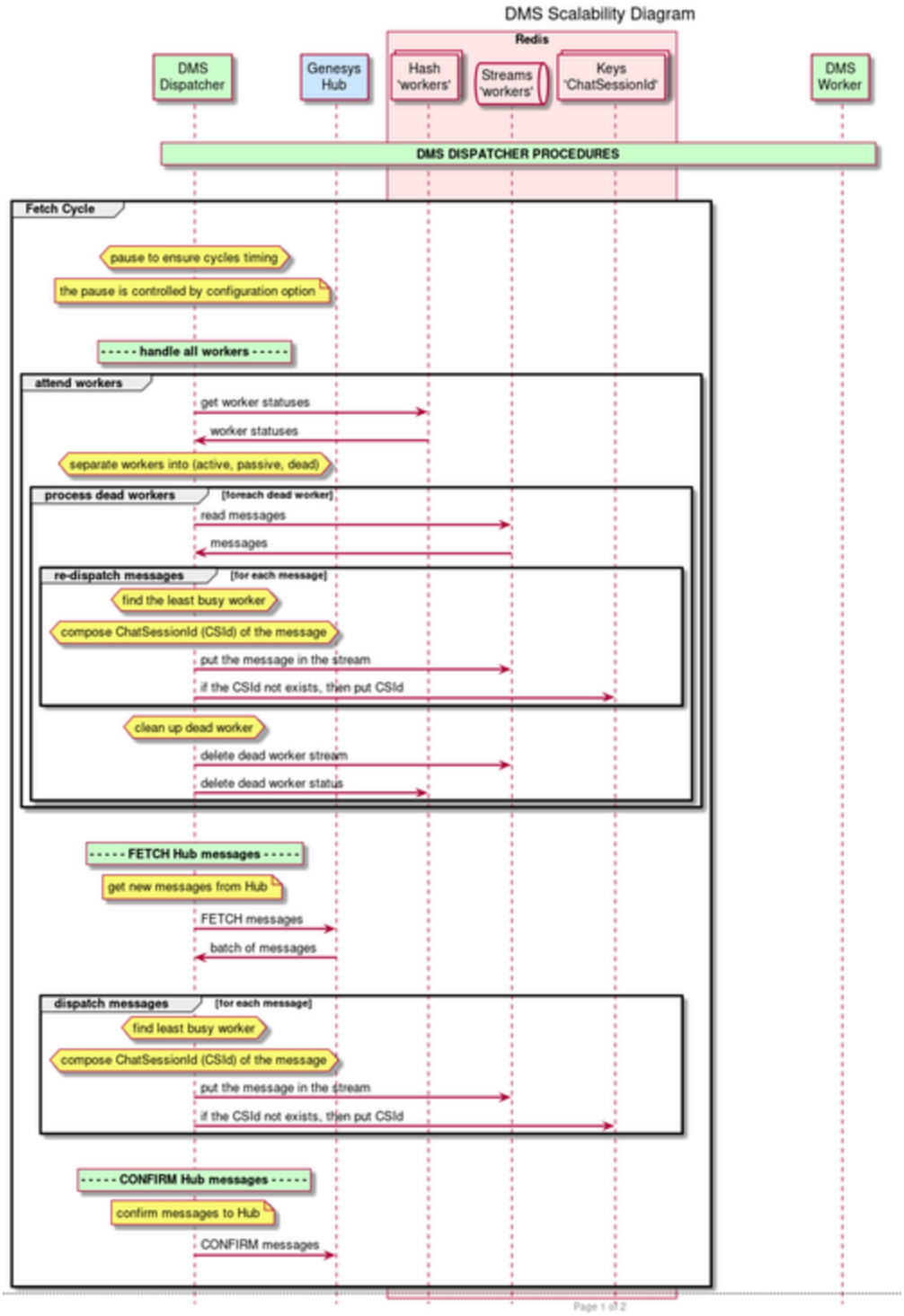
Architecture

> **Important**
>
> - The current implementation of DMS High Availability works without any additional changes. Each DMS-w relies on its backup to restore those established chat sessions in redundant Chat Servers.

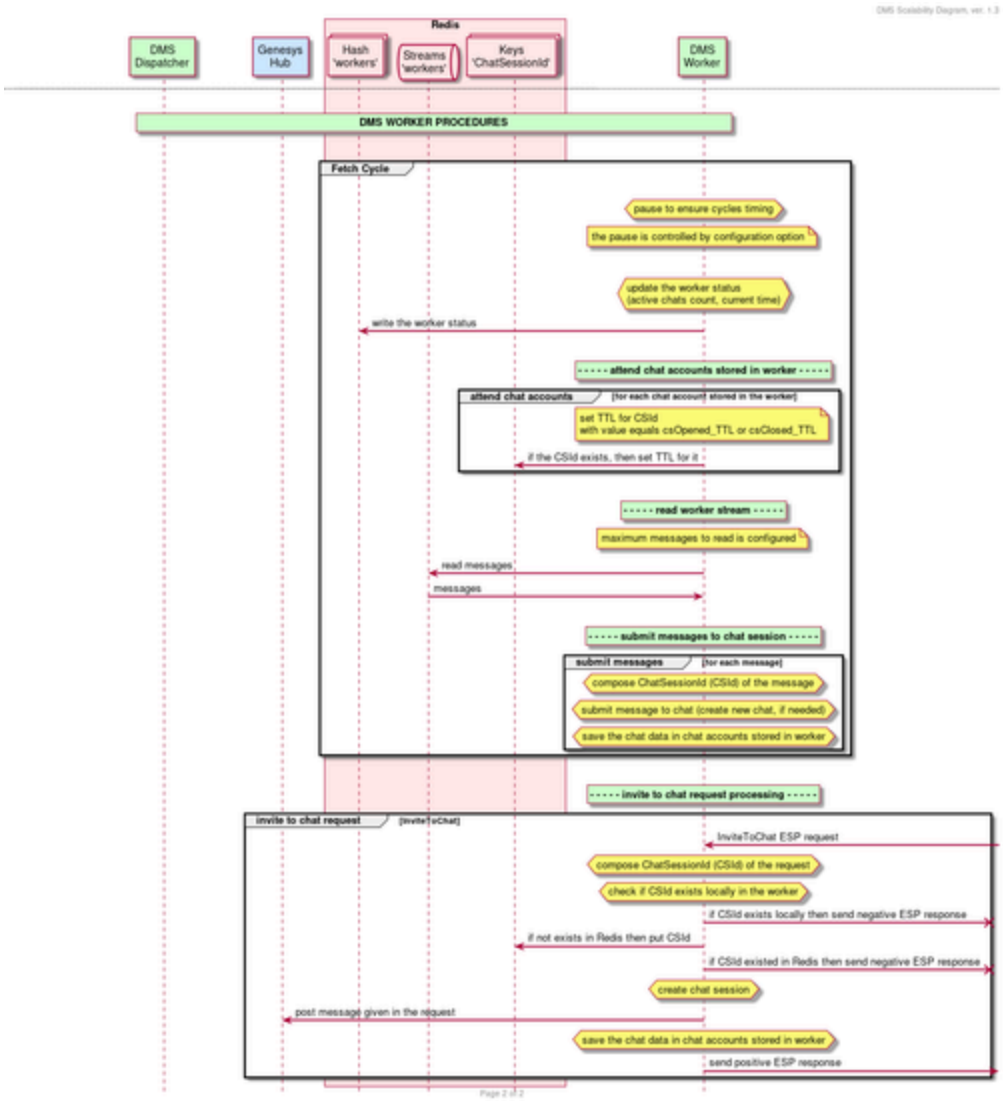
- The architecture diagram shown below do not represent backups.



Dispatcher workflow



Worker workflow



DMS-d responsibilities

- Gets authorization token similar to how a regular DMS does.
- Fetches messages from Genesys Hub and puts them into Redis.
- Confirms fetched messages to Hub after it saved them in Redis.
- Distributes messages among the available workers.
- Detects inactive workers to re-dispatch messages between active workers.

DMS-w responsibilities

- Gets authorization token similar to how a regular DMS does.
- Records its status in Redis regularly to indicate its state.
- Reads the messages assigned to it from Redis and creates chat sessions.
- Updates the records in Redis with the chat session IDs and its own name.
- Reads messages from Redis, places them into the existing chats, and marks them as processed in Redis so that if a session needs to be restored in another worker, then the dispatcher can identify which messages would require re-sending.
- Sends outbound messages to Genesys Hub directly.

Configuration

The following options in the **settings** section can be used to configure DMS scalability:

- The server-mode option indicates the mode of operation for DMS.
- The redis-client-config option specifies the file name of the Redis client configuration (.yaml file).

Redisson client configuration

Genesys recommends that you use a secured connection with the Redis server. To ensure the secured connection, the Redisson client configuration must specify the address of the Redis server in the form of `rediss:` . Refer to the example for a sample address.

Additionally, you must describe the JKS key store (parameter `sslTruststore`) with the CA certificate which is used to verify the Redis server's certificate. Example:

```
singleServerConfig:
 idleConnectionTimeout: 10000
 connectTimeout: 10000
 timeout: 3000
 retryAttempts: 3
 retryInterval: 1500
 password: "password"
 subscriptionsPerConnection: 5
 clientName: "DMS"
 sslEnableEndpointIdentification: true
 sslProvider: "JDK"
 # sslTruststore: "file:///C:/Redis_TLS/RedisCA_Test.keystore"
 pingConnectionInterval: 30000
 keepAlive: false
 tcpNoDelay: false
 address: "redis://127.0.0.1:6379"
 # address: "rediss://wsl-win10desktop:6379"
 subscriptionConnectionMinimumIdleSize: 1
 subscriptionConnectionPoolSize: 5
 connectionMinimumIdleSize: 5
 connectionPoolSize: 10
 database: 0
 dnsMonitoringInterval: 5000
```

```
threads: 16
nettyThreads: 32
#codec: !<org.redisson.client.codec.JsonJacksonCodec> {}
referenceEnabled: true
transportMode: "NIO"
lockWatchdogTimeout: 30000
reliableTopicWatchdogTimeout: 600000
keepPubSubOrder: true
useScriptCache: false
minCleanUpDelay: 5
maxCleanUpDelay: 1800
cleanUpKeysAmount: 100
nettyHook: !<org.redisson.client.DefaultNettyHook> {}
useThreadClassLoader: true
addressResolverGroupFactory: !<org.redisson.connection.DnsAddressResolverGroupFactory> {}
```

JVM parameters for DMS-d

```
-Dgenesys.mcr.stdserverex.apptype=CFGSocialMS
-Dgenesys.mcr.stdserverex.espserver=false
-Dgenesys.mcr.stdserverex.itxrequired=false
-Dgenesys.mcr.stdserverex.scsrequired=true
-Dgenesys.mcr.stdserverex.chatrequired=false
-Dgenesys.mcr.stdserverex.ucsrequired=false
-Dgenesys.mcr.stdserverex.ucsclusterrequired=false
-Dgenesys.mcr.stdserverex.espserver.multipleespports=false
-Dgenesys.mcr.stdserverex.lmsfile=dmserver.lms
-Dgenesys.mcr.stdserverex.infoproviderservice=com.genesyslab.mcr.smserver.OwnInfoProvider
-Dgenesys.mcr.stdserverex.flexchatprotocol=true
```

JVM parameters for DMS-w

```
-Dgenesys.mcr.stdserverex.apptype=CFGSocialMS
-Dgenesys.mcr.stdserverex.espserver=true
-Dgenesys.mcr.stdserverex.itxrequired=true
-Dgenesys.mcr.stdserverex.scsrequired=true
-Dgenesys.mcr.stdserverex.chatrequired=true
-Dgenesys.mcr.stdserverex.ucsrequired=true
-Dgenesys.mcr.stdserverex.ucsclusterrequired=false
-Dgenesys.mcr.stdserverex.espserver.multipleespports=false
-Dgenesys.mcr.stdserverex.lmsfile=dmserver.lms
-Dgenesys.mcr.stdserverex.infoproviderservice=com.genesyslab.mcr.smserver.OwnInfoProvider
-Dgenesys.mcr.stdserverex.flexchatprotocol=true
```

Configuration layer connections

You must ensure the following connections in the Configuration Layer:

DMS instance	Connects to
DMS-d	• Solution Control Server • Message Server, if required
DMS-w	• Chat Servers

DMS instance	Connects to
	• Contact Server or cluster • Interaction Server • Solution Control Server • Message Server, if required

(Optional) Genesys Hub Plug-in for Workspace Desktop Edition configuration

You can specify multiple DMS servers to be used for DMS scalability. If the DMS app name is not specified in the following section, DMS app name from user data of a parent interaction is used.

> **Important**
>
> Genesys Hub Plug-in for Workspace Desktop Edition 9.1.007.07 or later is required for configuring the following sections.

Configure the following sections in the WDE application for the corresponding media channels:

Section	Channel	Value	Changes take effect	Type
applebcsession.appname	Channel name for Apple Business Chat	DMS app name. Multiple servers can be specified by using comma.	Immediately	String
whatsappsession.appname	Channel name for WhatsApp	DMS app name. Multiple servers can be specified by using comma.	Immediately	String