

MigratoryData Server

Architecture Guide

Version 6.0
July 12, 2018



Copyright Information

Copyright © 2007-2018 Migratory Data Systems. ALL RIGHTS RESERVED.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE DOCUMENT. MIGRATORY DATA SYSTEMS MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT DESCRIBED IN THIS DOCUMENT AT ANY TIME.

Contents

1	Introduction	3
1.1	Release	3
1.2	Related Documents	3
2	Product Overview	4
2.1	Introducing MigratoryData	5
2.2	Inside MigratoryData	5
2.3	MigratoryData Features	5
2.4	MigratoryData Client Libraries	9
2.5	MigratoryData Extension Libraries	9
2.6	Supported Platforms	10

1. Introduction

This guide provides an overview of MigratoryData Server and explains its concepts and features.

1.1 Release

This guide is part of the documentation set for MigratoryData Server version 6.0.

1.2 Related Documents

- *MigratoryData Installation Guide*
- *MigratoryData Configuration Guide*
- *MigratoryData Performance Benchmarking Guide*
- *Developer's Guide* and *Reference Manual* of any available client library
- *Developer's Guide* and *Reference Manual* of any available extension library

2. Product Overview

This chapter introduces MigratoryData Server.

Topics

2.1	Introducing MigratoryData	5
2.2	Inside MigratoryData	5
2.3	MigratoryData Features	5
2.4	MigratoryData Client Libraries	9
2.5	MigratoryData Extension Libraries	9
2.6	Supported Platforms	10

2.1 Introducing MigratoryData

MigratoryData is a truly scalable messaging system able to push real-time messages to millions of connected users and devices across Web, Mobile, and IoT.

MigratoryData is the only messaging system written to address the C1M problem, and even the C10M problem, depending on the volume of messages required. One instance of MigratoryData, running a commodity server, has been benchmarked to handle 1 million and respectively 10 million concurrent clients, while pushing substantial real-time messaging traffic, of the order of 1 Gbps.

Since a decade, MigratoryData has been powering large real-time messaging services having hundreds of millions of users.

2.2 Inside MigratoryData

Written for architects, the paper "Reliable Messaging to Millions of Users with MigratoryData" presents the inner workings of MigratoryData and how high scalability is achieved without sacrificing reliability.

This paper was originally presented at the ACM/IFIP/USENIX International Middleware Conference 2017, University of Nevada, Las Vegas and published in the proceedings of Middleware 2017, copyright ACM, 2017. A preprint of the paper is available from arXiv at:

<https://arxiv.org/abs/1712.09876>

2.3 MigratoryData Features

This section describes the main functionalities of MigratoryData.

Universal Pub/Sub Messaging

MigratoryData provides a topic-based publish/subscribe Application Programming Interface (API) with libraries for the the most popular technologies. MigratoryData's libraries for JavaScript, iOS, Android, Java, .NET, C++, PHP, Python, and Ruby can be used to build a wide range of real-time applications across web, mobile, desktop, and server.

The WebSocket protocol (open IETF standard) is used to connect to the MigratoryData server the web clients built with MigratoryData's client library for JavaScript and running on modern web browsers. Web clients running on old web browsers, without support for WebSockets, as well as all other mobile, desktop, or server clients built with MigratoryData's libraries use the HTTP protocol (open IETF standard) along persistent connections to connect to the MigratoryData

server. In this way, behind the scene, every MigratoryData client connects to the MigratoryData server along a persistent TCP connection, enabling low-latency messaging.

Messaging for IoT Devices

MigratoryData provides support for the MQTT protocol (open ISO Standard), the de-facto protocol and standard for the Internet of Things (IoT). Thus, MigratoryData can be used as a scalable MQTT broker for IoT devices.

In addition, there are many MQTT libraries available with very active community such as Eclipse's PAHO libraries. These libraries can be used to build real-time applications with technologies not currently supported by MigratoryData's libraries. MigratoryData's real-time protocol implemented by MigratoryData's library is very bandwidth efficient. However, being a text protocol, it is slightly less bandwidth efficient than MQTT which is a binary protocol. So, MQTT libraries might be preferable over MigratoryData's library if extreme bandwidth optimization is required or even if an open-standards policy applies.

Note that MigratoryData only supports QoS level 0 for MQTT at this time. MigratoryData's libraries might be used if the higher QoS levels 1 or 2 are required.

High Scalability

MigratoryData takes full advantage of all CPU cores available on the machine on which it is installed, providing an unparalleled vertical scalability. It is the only messaging system written to address the C1M problem, and even the C10M problem, depending on the volume of messages required. A single instance of MigratoryData, running a commodity server, has been benchmarked to handle 1 million and respectively 10 million concurrent clients, while pushing substantial real-time messaging traffic, of the order of 1 Gbps.

In addition, multiple MigratoryData instances can be clustered to scale horizontally. MigratoryData uses subscribers-partitioning (rather than topics-partitioning). The cluster members being independent in terms of subscribers, MigratoryData scales linearly with the number of subscribers.

The high vertical scalability and the linear horizontal scalability in terms of subscribers make MigratoryData an unbeatable messaging solution for subscribers-dominant use cases, being able to handle any number of users and message volumes in a very cost-effective way.

Guaranteed Message Delivery

Multiple instances of MigratoryData can be deployed as a fault-tolerant cluster with no single point of failure.

Subscribers are guaranteed to receive all messages published to a topic, in the same order. MigratoryData assigns sequence numbers to incoming messages for a topic. This order respects the order of acknowledged publications sent by individual publishers (but messages from different publishers may be ordered arbitrarily). Each server maintains a cache of messages for all topics. A subscriber can detect and ask for missed messages upon a reconnection using these sequence numbers.

MigratoryData adopts replication to synchronize the cache of messages among cluster members. The fault-tolerant replication algorithm is detailed in the paper mentioned in Section 2.2.

Security

MigratoryData is used in large enterprise deployments, including at a number of Fortune Global 500 companies, where highest security requirements need to be met. Thus, MigratoryData uses the security standards like TLS/SSL to encrypt any access either from clients or administrators. Specific TLS/SSL ciphers can be enabled or disabled.

Various authorization mechanisms exist, including the possibility to build custom authorization rules with MigratoryData's Authorization Extension API and plug them into MigratoryData, to allow or deny each user to subscribe to or publish messages on certain topics. In addition, inter-cluster communication uses authentication.

MigratoryData can be deployed in the Demilitarized Zone (DMZ) to take full advantage of the dual-firewall topology, where the internal firewall may close all ports for incoming connections. Finally, MigratoryData can be configured to run as a normal (non-root) user.

Advanced Monitoring

MigratoryData provides various monitoring metrics. The access to the monitoring metrics is secure using encrypted connections and authentication. Also, it is standards-based using the Java Management Extensions (JMX) and HTTP standards. There are many utilities such as Oracle's `jconsole` for JMX or various RRDTool-based solutions for HTTP for graphical monitoring.

Various metrics are available such as the number of connected clients, the connection/disconnection rate, the incoming/outgoing throughput of messages. For each metrics, various statistics exist like max/mean/stdev for various time periods like last 1, 5, 15 minutes/hours/days/months.

In addition, all monitoring metrics and statistics can be obtained with MigratoryData's Audit Extension API as part of a custom extension which can be plugged into MigratoryData.

Capacity Planning

MigratoryData provides sophisticated benchmarking tools, able to simulate millions of concurrent clients efficiently and being able to publish messages of a configurable size at a configurable frequency. These tools can be used to estimate the hardware and the number of instances required by a real-time service to assess its feasibility or before moving it to production.

2.4 MigratoryData Client Libraries

MigratoryData provides one common publish/subscribe API with libraries for the languages and technologies listed in Table 2.1 below.

Application Type	Client Library	Functions
Web Applications	JavaScript	publish and subscribe
Mobile Applications	iOS	publish and subscribe
	Android	publish and subscribe
Desktop/Server Applications	Java	publish and subscribe
	.NET	publish and subscribe
	C++	publish and subscribe
	PHP	publish
	Python	publish
	Ruby	publish

Table 2.1: MigratoryData's Client Libraries

In addition, MigratoryData provides support for the MQTT protocol¹. The many MQTT client libraries available with very active community – such as Eclipse's PAHO libraries – can be used to build applications for the Internet of Things (IoT).

2.5 MigratoryData Extension Libraries

MigratoryData can be extended with custom plugins. Two easy-to-use Java API libraries can be used to build extensions which can be plugged into the MigratoryData server. Table 2.2 below details these extension libraries.

¹MigratoryData only supports QoS level 0 for MQTT at this time. MigratoryData's client libraries might be used if the higher QoS levels 1 or 2 are required.

Extension Library	Functions
Authorization	<p>Verifies if a user is authorized to connect to the server</p> <p>Verifies if a user is authorized to subscribe to certain topics</p> <p>Verifies if a user is authorized to publish on certain topics</p>
Audit	<p>Provides access events such as connect, disconnect, subscribe, unsubscribe</p> <p>Provides stats events such as number of clients, connection / disconnection rate, incoming / outgoing message throughput</p> <p>Provides message events consisting in any new message accepted by the server</p> <p>Provides cache events whenever a message is added to or accessed or deleted from the cache</p>

Table 2.2: MigratoryData's Extensions Libraries

2.6 Supported Platforms

Written in pure Java, the MigratoryData server runs on all major operating systems. While running the MigratoryData server on operating systems such as Windows or OS X could be convenient for development, for production deployments, the recommended operating system is Linux; more precisely, any modern 64-bit Linux enterprise distribution (kernel 2.6 or later) such as RedHat Enterprise Linux, CentOS, or Debian version 6 or later.

MigratoryData Server is self-contained, its object code consists of a single JAR file with no dependencies. Also, the installers for Linux are bundled with Oracle's Java Runtime Environment (JRE) version 8 such that the Linux installers have no dependencies. On the other hand, the platform-independent tarball installer requires Oracle JRE version 8 or later that needs to be installed separately.

