



Design of a Distributed Messaging Queue: Part 1

Learn about the high-level design of a messaging queue and how to scale the metadata of queues.

We'll cover the following

- Distributed messaging queue
 - High-level design
 - Load balancer
 - Front-end service
 - Metadata service

So far, we've discussed the requirements and design considerations of a distributed messaging queue. Now, let's begin with learning about the high-level design of a distributed messaging queue.

Distributed messaging queue

Unlike a single-server messaging queue, a distributed messaging queue resides on multiple servers. A distributed messaging queue has its own challenges. However, it resolves the drawbacks of a single-server messaging queue if designed properly.


The following sections focus on the scalability, availability, and durability issues of designing a distributed messaging queue by introducing us to a more fault-tolerant architecture of a messaging queue.



High-level design

Before diving deep into the design, let's assume the following points to make the discussion more simple and easy to understand. In the upcoming material, we discuss how the following assumptions enable us to eliminate the problems in a single-server solution to a messaging queue.



- 
1. Queue data is replicated using either a primary-secondary or quorum-like system inside a cluster (read through the [Data Replication](#) lesson for more details). Our service can use data partitioning if the queue gets too long to fit on a server. We can use a consistent hashing-like scheme for this purpose, or we may use a key-value store where the key might be the sequence numbers of the messages. In that case, each shard is appropriately replicated (refer to the [Partition](#) lesson for more details on this).
 2. We also assume that our system can auto-expand and auto-shrink the resources as per the need to optimally utilize resources.

The following figure demonstrates a high-level design of a distributed messaging queue that's composed of several components.

