

图1. 消息队列组模型

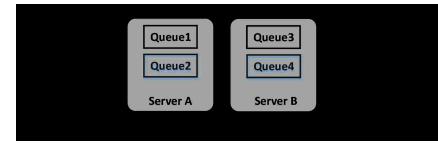


图2. 集群战略图

### A. 集群战略

大多数消息传递中间件产品都支持消息服务器的集群，以确保消息源的可靠性。群集中的每个消息传递服务器都有一个具有详细消息分发功能的单独消息队列。现有的聚类方案分为高性能和高可用性。卓越的性能方案旨在提供统一服务时间内的更高吞吐量。高可用性方案旨在减少整个消息服务的中断时间。在SIS集成应用程序环境中，消息服务设计的主要目标是确保服务的正常运行并提供高度可靠的灾难恢复性能。现有的消息中间件产品[15-17]提供了各种高可用性集群策略，主要分为两部分：服务处理节点备份和数据备份。服务处理节点备份用于备份消息服务器中的消息转发功能。数据备份是消息队列中消息的备份。不同的集群策略可以在不同的应用场景中使用。在SIS中，服务处理节点备份和数据备份同样重要。但是，备份策略与群集中硬件的大小密切相关。考虑到离岸环境中硬件设备的有限性能，对服务处理节点和数据使用完全备份是不可接受的。为了保证服务节点的容灾能力，我们希望服务处理节点使用完整的备份策略。另一方面，消息队列数据使用不完整的备份策略来减少数据的冗余。结合消息队列组模型，我们提出了一个新的集群策略，如图4所示。消息队列组中的多个消息队列实体具有相同的消息数据，实现数据的冗余备份。同时，为了减少数据的冗余，消息队列组中的消息队列实体的数量 $N_q \leq N_s$ ，并分布在不同的消息服务器中。当 $N_s=1$ 或 $2$ 时， $N_q=N_s$ 。当 $N_s \geq 3$ ， $N_q \leq N_s$ 时，根据消息队列组中的消息加载条件调整 $N_q$ 。

### B. 消息服务器监控模块

为了监控消息服务器和消息队列的实时状态，我们设计了消息服务器监控模块。监控模块独立于每个消息服务器运行，以获取每个消息服务器节点的各种信息，包括系统状态，AMQ代理状态和消息队列状态。监控模块主要分为JavaManagementExtensions (JMX) 管理模块，Sigar模块和Dubbo模块。如图5所示。

其中，JMX管理模块使用AMQ提供的JMExtension接口来获取和控制Broker和消息队列的运行状态信息。Sigar是HypericHQ的主要数据收集组件，可用于从多个操作系统收集信息[18]。我们使用Sigar工具获取系统状态信息，如CPU，磁盘，网络I/O，内存和JVM。Dubbo是一个基于Java的远程过程调用 (RPC) 框架，它提供基于接口的远程调用和自动服务注册和发现[19]。

监控模块使用Dubbo为配置中心提供远程监控和控制接口。配置中心使用这些接口统一管理消息服务器和消息队列。另一方面，使用Dubbo的自动服务注册和发现功能用于实现消息服务器的灵活扩展。