

# 微服务治理的挑战（分布式数据管理）

作者: [hsxian](#)

原文链接: <https://ld246.com/article/1555643275867>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



本文参照MSDN的[分布式数据管理的挑战和解决方案](#)而写，更多信息，请访问原文。

## 1. 如何定义每个微服务的边界

微服务要求高内聚，低耦合的理念，势必带来微服务边界划分的难题。那么，如何识别这些边界呢？

边界以内为服务领域，划分不同的服务也就对应着不同的域划分。意即，域的概念模型因子系统或服务而异。此原则类似于[域驱动开发\(DDD\)](#)，其中每界限上下文([bounded context](#))或者自治子系统或服务必须拥有其域模型（数据加上逻辑和行为）。每个 DDD 界定上下文对应于一个业务微服务（一个或服务）。所以说，DDD是实现微服务的一种途径，而不是唯一途径。

## 2. 如何创建从多个微服务中检索数据的查询

每个微服务只负责其边界内部的数据查询任务。但不可避免的，我们的应用可能检索多个服务的数据由此而引发多次调用不同服务的状况。这是一种低效率的查询行为。如要提高系统通信效率，那么我需要通过某种方法来聚合信息。

**API 网关。**用网关来聚合多个数据查询任务，然后作为整体返回给调用者。但API 网关也存在着诸多险，比如：

- 实现API网关，需要与微服务进行耦合，造成了侵入。
- 单个API网关可能存在性能瓶颈或单点故障。并可能违反微服务自治原则，若要降低这种可能性，以采用多个细粒度的API网关。
- API网关会增加而外的开发与运维成本。
- 如果是由一个团队开发的 API 网关，则可能出现开发瓶颈。

**具有查询/读取表的 CQRS。**此方法要求生成一个包含多个微服务的数据的只读表，以此针对复杂的个数据模块查询。该表字段可以和前端所需字段一一对应。该方法不仅解决了多数据查询挑战，还会来显著的性能提升。但是该方法也具有一些而外的挑战，比如，额外的开发工作；最终一致性的达成

为达到高性能和高可伸缩，要开发多个数据库的CQRS。

**中央数据库的“冷数据”**。对于实时性要求低的查询任务，可以将“热数据”（来自微服务）作为“冷数据”导出到用于报告的大型数据库中（如Hadoop）。数据来源更新和事务必须位于微服务。同步数的方法可以采用：事件驱动的通信；数据导入/导出工具。

在任何情况下，都需要通过强大的依赖项、内聚和数据聚合在各微服务的发展自治和部署之间取得平衡。

### 3. 如何实现微服务之间的一致性

由于多个微服务之间彼此边界隔离，因此，保持多个微服务之间的一致性的同时实现端到端的业务成一项挑战。

建议使用[异步事件驱动通信](#)。

### 4. 如何设计跨微服务边界的通信

跨微服务边界的通信是真正的挑战。在此上下文中，通信不是指应使用何种协议（HTTP 和 REST、AQP、消息等）。相反，它解决了应使用何种通信方式的问题，尤其是应如何耦合微服务。

为强化微服务自治并且提升复原能力，应尽可能少地使用跨微服务的请求/响应通信链。建议只使用异步交互进行微服务之间的通信，方法是使用基于消息和基于事件的异步通信，或（异步）使用独立于始 HTTP 请求/响应周期的 HTTP 轮询。

异步通信的使用稍后将在本指南的[异步微服务集成强化微服务的自治性](#)和[基于消息的异步通信](#)部分详细介绍。