第一部分 数据系统基础

前四章主要讲述数据系统的基础知识，适用于单机或分布式集群。

1. 第一章，主要介绍本书使用的术语和方法，阐述了可靠性、扩展性和易维护性等系统指标的含义，以及实现以上目标的方式方法。
2. 第二章，通过比较几种不同的数据模型和查询语言（不同数据库之间最为明显的区别）。我们可以得出不同场景下最合适的数据模型
3. 第三章，研究存储引擎内部原理，了解数据如何落地磁盘。不同的存储引擎适用于不同的服务负载，选择合适的存储引擎可以实现最佳性能。
4. 第四章，对比数据序列化的方式，特别强调为适应系统需求改变或计划调整，导致序列化方式高昂的代价。

# 第1章 系统的可靠性、扩展性和维护性

当今互联网时代，很多系统是数据密集型的，与之相对是计算密集型的。CPU很难成为这些系统的瓶颈，而数据的体量、复杂性以及变化频率却是非常棘手的问题。

通常，构建数据密集型应用，需要考虑以下几个方面：

1. 存储数据，便于查询（数据库）；
2. 热数据缓存化，加快查询速度（缓存）；
3. 提供搜索功能（检索）；
4. 异步化处理消息（流式处理）；
5. 周期性处理大量累计数据（批处理）；

如果系统具备以上功能，说明系统抽象非常完美。但是，工程师构建系统时，不会刻意设计一套新的存储引擎，因为主流的数据库已经可以完美的解决大部分基本需求。

实际上却不是这样的，不同的数据库有不同的特性，每个系统有各自的需求。拿构建缓存、索引来讲，有很多种有效方式。工程师的工作应该是评估哪些组件更适合当前的系统，但选型与集成这些组件来实现系统设计是比较困难的。

本书会结合理论和实践两方面，在构建数据密集型系统方面，给大家提供建议。我们会阐述组件之间的共性、特性以及其实现方式。

本章节，我们会了解数据系统可靠性、扩展性、维护性等方面的基本原理。我们将会阐明以上特性的含义，概括几种思考方式，以及复习基础知识。接下来的章节，我们会聚焦设计数据密集型系统的选型和决策。

## 数据系统的思考

通常，我们认为数据库、队列、缓存等组件是完全不同的组件。虽然数据库和队列有很多相似点——可以存储数据一段时间，但是二者有不同的访问模式，这就导致了不同的性能特性和实现方式。

但是，为什么我们应该把二者集成在数据系统内呢？