第1章 概 述

Google、Amazon、Alibaba等互联网公司的成功催生了云计算和大数据两大热门领域。无论是云计算、大数据还是互联网公司的各种应用，其后台基础设施的主要目标都是构建低成本、高性能、可扩展、易用的分布式存储系统。

虽然分布式系统研究了很多年，但是，直到近年来，互联网大数据应用的兴起才使得它大规模地应用到工程实践中。相比传统的分布式系统，互联网公司的分布式系统具有两个特点：一个特点是规模大，另一个特点是成本低。不同的需求造就了不同的设计方案，可以这么说，Google等互联网公司重新定义了大规模分布式系统。本章介绍大规模分布式系统的定义与分类。

1.1 分布式存储概念

大规模分布式存储系统的定义如下：

“分布式存储系统是大量普通PC服务器通过Internet互联，对外作为一个整体提供存储服务。”

分布式存储系统具有如下几个特性：

- 可扩展。分布式存储系统可以扩展到几百台甚至几千台的集群规模，而且，随着集群规模的增长，系统整体性能表现为线性增长。
- 低成本。分布式存储系统的自动容错、自动负载均衡机制使其可以构建在普通PC机之上。另外，线性扩展能力也使得增加、减少机器非常方便，可以实现自动运维。
- 高性能。无论是针对整个集群还是单台服务器，都要求分布式存储系统具备高性能。
- 易用。分布式存储系统需要能够提供易用的对外接口，另外，也要求具备完善的监控、运维工具，并能够方便地与其他系统集成，例如，从Hadoop云计算系统导入数据。

分布式存储系统的挑战主要在于数据、状态信息的持久化，要求在自动迁移、自动容错、并发读写的过程中保证数据的一致性。分布式存储涉及的技术主要来自两个领域：分布式系统以及数据库，如下所示：

- 数据分布：如何将数据分布到多台服务器才能够保证数据分布均匀？数据分布到多台服务器后如何实现跨服务器读写操作？
- 一致性：如何将数据的多个副本复制到多台服务器，即使在异常情况下，也能够保证不同副本之间的数据一致性？
- 容错：如何检测到服务器故障？如何自动将出现故障的服务器上的数据和服务迁移到集群中其他服务器？
- 负载均衡：新增服务器和集群正常运行过程中如何实现自动负载均衡？数据迁移的过程中如何保证不影响已有服务？
- 事务与并发控制：如何实现分布式事务？如何实现多版本并发控制？
- 易用性：如何设计对外接口使得系统容易使用？如何设计监控系统并将系统的内部状态以方便的形式暴露给运维人员？
- 压缩 / 解压缩：如何根据数据的特点设计合理的压缩 / 解压缩算法？如何平衡压缩算法节省的存储空间和消耗的 CPU 计算资源？

分布式存储系统挑战大，研发周期长，涉及的知识面广。一般来讲，工程师如果能够深入理解分布式存储系统，理解其他互联网后台架构不会再有任何困难。

1.2 分布式存储分类

分布式存储面临的数据需求比较复杂，大致可以分为三类：

- 非结构化数据：包括所有格式的办公文档、文本、图片、图像、音频和视频信息等。
- 结构化数据：一般存储在关系数据库中，可以用二维关系表结构来表示。结构化数据的模式（Schema，包括属性、数据类型以及数据之间的联系）和内容是分开的，数据的模式需要预先定义。
- 半结构化数据：介于非结构化数据和结构化数据之间，HTML 文档就属于半结构化数据。它一般是自描述的，与结构化数据最大的区别在于，半结构化数据的模式结构和内容混在一起，没有明显的区分，也不需要预先定义数据的模式结构。

不同的分布式存储系统适合处理不同类型的数据，本书将分布式存储系统分为四类：分布式文件系统、分布式键值（Key-Value）系统、分布式表格系统和分布式数据库。

1. 分布式文件系统

互联网应用需要存储大量的图片、照片、视频等非结构化数据对象，这类数据以对象的形式组织，对象之间没有关联，这样的数据一般称为 Blob（Binary Large Object，二进制大对象）数据。

分布式文件系统用于存储 Blob 对象，典型的系统有 Facebook Haystack 以及 Taobao File System（TFS）。另外，分布式文件系统也常作为分布式表格系统以及分布式数据库的底层存储，如谷歌的 GFS（Google File System，存储大文件）可以作为分布式表格系统 Google Bigtable 的底层存储，Amazon 的 EBS（Elastic Block Store，弹性块存储）系统可以作为分布式数据库（Amazon RDS）的底层存储。

总体上看，分布式文件系统存储三种类型的数据：Blob 对象、定长块以及大文件。在系统实现层面，分布式文件系统内部按照数据块（chunk）来组织数据，每个数据块的大小大致相同，每个数据块可以包含多个 Blob 对象或者定长块，一个大文件也可以拆分为多个数据块，如图 1-1 所示。分布式文件系统将这些数据块分散到存储集群，处理数据复制、一致性、负载均衡、容错等分布式系统难题，并将用户对 Blob 对象、定长块以及大文件的操作映射为对底层数据块的操作。

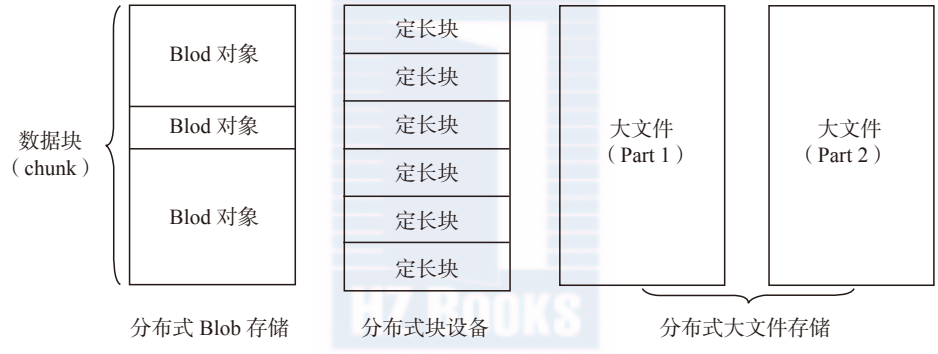

图 1-1 数据块与 Blob 对象、定长块、大文件之间的关系

2. 分布式键值系统

分布式键值系统用于存储关系简单的半结构化数据，它只提供基于主键的 CRUD（Create/Read/Update/Delete）功能，即根据主键创建、读取、更新或者删除一条键值记录。

典型的系统有 Amazon Dynamo 以及 Taobao Tair。从数据结构的角度看，分布式键值系统与传统的哈希表比较类似，不同的是，分布式键值系统支持将数据分布到集群中的多个存储节点。分布式键值系统是分布式表格系统的一种简化实现，一般用作缓存，比如淘宝 Tair 以及 Memcache。一致性哈希是分布式键值系统中常用的数据分布技术，因其被 Amazon DynamoDB 系统使用而变得相当有名。

3. 分布式表格系统

分布式表格系统用于存储关系较为复杂的半结构化数据，与分布式键值系统相比，分布式表格系统不仅仅支持简单的 CRUD 操作，而且支持扫描某个主键范围。分布式

表格系统以表格为单位组织数据，每个表格包括很多行，通过主键标识一行，支持根据主键的 CRUD 功能以及范围查找功能。

分布式表格系统借鉴了很多关系数据库的技术，例如支持某种程度上的事务，比如单行事务，某个实体组（Entity Group，一个用户下的所有数据往往构成一个实体组）下的多行事务。典型的系统包括 Google Bigtable 以及 Megastore，Microsoft Azure Table Storage，Amazon DynamoDB 等。与分布式数据库相比，分布式表格系统主要支持针对单张表格的操作，不支持一些特别复杂的操作，比如多表关联，多表联接，嵌套子查询；另外，在分布式表格系统中，同一个表格的多个数据行也不要求包含相同类型的列，适合半结构化数据。分布式表格系统是一种很好的权衡，这类系统可以做到超大规模，而且支持较多的功能，但实现往往比较复杂，而且有一定的使用门槛。

4. 分布式数据库

分布式数据库一般是从单机关系数据库扩展而来，用于存储结构化数据。分布式数据库采用二维表格组织数据，提供 SQL 关系查询语言，支持多表关联，嵌套子查询等复杂操作，并提供数据库事务以及并发控制。

典型的系统包括 MySQL 数据库分片（MySQL Sharding）集群，Amazon RDS 以及 Microsoft SQL Azure。分布式数据库支持的功能最为丰富，符合用户使用习惯，但可扩展性往往受到限制。当然，这一点并不是绝对的。Google Spanner 系统是一个支持多数据中心的分布式数据库，它不仅支持丰富的关系数据库功能，还能扩展到多个数据中心的成千上万台机器。除此之外，阿里巴巴 OceanBase 系统也是一个支持自动扩展的分布式关系数据库。

关系数据库是目前为止最为成熟的存储技术，它的功能极其丰富，产生了商业的关系数据库软件（例如 Oracle，Microsoft SQL Server，IBM DB2，MySQL）以及上层的工具及应用软件生态链。然而，关系数据库在可扩展性上面临着巨大的挑战。传统关系数据库的事务以及二维关系模型很难高效地扩展到多个存储节点上，另外，关系数据库对于要求高并发的应用在性能上优化空间较大。为了解决关系数据库面临的可扩展性、高并发以及性能方面的问题，各种各样的非关系数据库风起云涌，这类系统成为 NoSQL 系统，可以理解为“Not Only SQL”系统。NoSQL 系统多得让人眼花缭乱，每个系统都有自己的独到之处，适合解决某种特定的问题。这些系统变化很快，本书不会尝试去探寻某种 NoSQL 系统的实现，而是从分布式存储技术的角度探寻大规模存储系统背后的原理。