第 2 章

飞天开放平台总览

本章简要介绍飞天开放平台，包括飞天平台体系架构、飞天平台内核和构建在飞天平台内核上的飞天开放服务。2.1 节介绍飞天平台的体系架构；接下来 2.2 节介绍飞天平台内核中各个模块的功能和设计，包括分布式系统底层服务、分布式文件系统、任务调度、集群监控和部署；最后 2.3 节简要介绍飞天平台开放服务，对各个开放服务的详细描述安排在第 3 章到第 7 章中。

2.1 飞天平台体系架构

如图 2.1 所示是飞天平台的体系架构图。整个飞天平台包括飞天内核（图 2.1 中浅灰色组件）和飞天开放服务（图 2.1 中白色组件）两大部分。飞天内核为上层的飞天开放服务提供存储、计算和调度等方面的底层支持，对应于图 2.1 中的协调服务、远程过程调用、安全管理、资源管理、分布式文件系统、任务调度、集群部署和集群监控模块。飞天开放服务为用户应用程序提供了存储和计算两方面的接口和服务，包括弹性计算服务（Elastic Compute Service，ECS）、开放存储服务（Open Storage Service，OSS）、开放结构化数据服务（Open Table Service，OTS）、关系型数据库服务（Relational Database Service，RDS）和开放数据处理服务（Open Data Processing Service，ODPS），并基于弹性计算服务提供了云服务引擎（Aliyun Cloud Engine，ACE）作为第三方应用开发和 Web 应用运行和托管的平台。

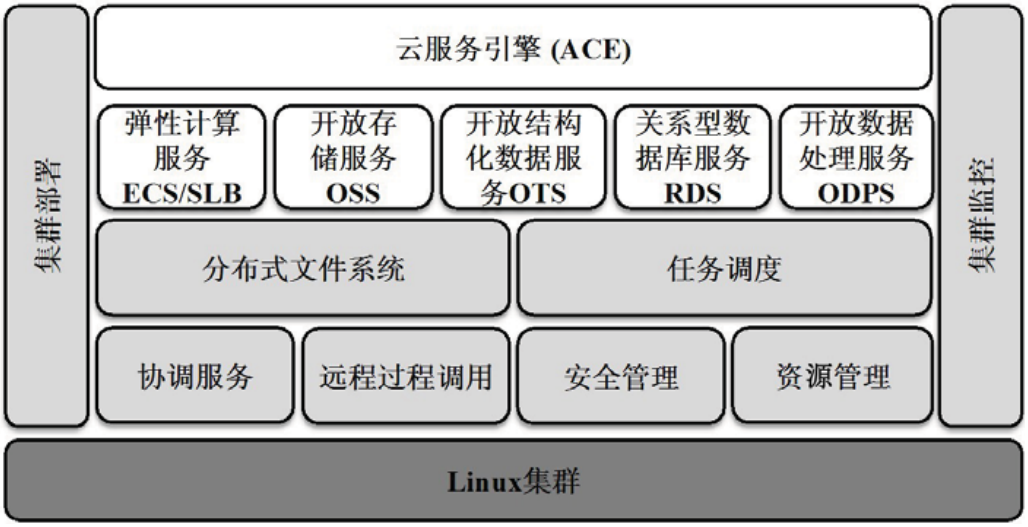

图 2.1　飞天平台的体系架构图

2.2　飞天平台内核

飞天平台内核包含的模块可以分为以下几部分。

- **分布式系统底层服务：**提供分布式环境下所需要的协调服务、远程过程调用、安全管理和资源管理的服务。这些底层服务为上层的分布式文件系统、任务调度等模块提供支持。
- **分布式文件系统：**提供一个海量的、可靠的、可扩展的数据存储服务，将集群中各个节点的存储能力聚集起来，并能够自动屏蔽软硬件故障，为用户提供不间断的数据访问服务；支持增量扩容和数据的自动平衡，提供类似于 POSIX 的用户空间文件访问 API，支持随机读写和追加写的操作。
- **任务调度：**为集群系统中的任务提供调度服务，同时支持强调响应速度的在线服务（Online Service）和强调处理数据吞吐量的离线任务（Batch Processing Job）；自动检测系统中故障和热点，通过错误重试、针对长尾作业并发备份作业等方式，保证作业稳定可靠地完成。
- **集群监控和部署：**对集群的状态和上层应用服务的运行状态和性能指标进行监控，对异常事件产生警报和记录；为运维人员提供整个飞天平台以及上层应用的部署和配置管理，支持在线集群扩容、缩容和应用服务的在线升级。

2.2.1　分布式系统底层服务

1．协调服务（女娲）

女娲（Nuwa）系统为飞天提供**高可用的协调服务（Coordination Service）**，是构建各

类分布式应用的核心服务，它的作用是采用类似文件系统的树形命名空间来让分布式进程互相协同工作。例如，当集群变更导致特定的服务被迫改变物理运行位置时，如服务器或者网络故障、配置调整或者扩容时，借助女娲系统可以使其他程序快速定位到该服务新的接入点，从而保证了整个平台的高可靠性和高可用性。

女娲系统基于类 Paxos 协议[15]，由多个女娲 Server 以类似文件系统的树形结构存储数据，提供高可用、高并发用户请求的处理能力。

女娲系统的目录表示一个包含文件的集合。与 UNIX 中的文件路径一样，女娲中路径是以“/”分割的，根目录（Root entry）的名字是“/”，所有目录的名字都是以“/”结尾的。与 UNIX 文件路径不同之处在于：女娲系统中所有文件或目录都必须使用从根目录开始的绝对路径。由于女娲系统的设计目的是提供协调服务，而不是存储大量数据，所以每个文件的内容（Value）的大小被限制在 1MB 以内。在女娲系统中，每个文件或目录都保存有创建者的信息。一旦某个路径被用户创建，其他用户就可以访问和修改这个路径的值（即文件内容或目录包含的文件名）。

女娲系统支持 Publish/Subscribe 模式，其中一个发布者、多个订阅者（One Publisher/Many Subscriber）的模式提供了基本的订阅功能；另外，还可用通过多个发布者、多个订阅者（Many Publisher/Many Subscriber）的方式提供分布式选举（Distributed Election）和分布式锁的功能。

再举一个使用女娲系统来实现负载均衡的例子：提供某一服务的多个节点，在服务启动的时候在女娲系统的同一目录下创建文件，例如，server1 创建文件“nuwa://cluster/myservice/server1”，server2 在同一目录下创建“nuwa://cluster/myservice/server2”。当客户端使用远程过程调用时，首先列举女娲系统服务中“nuwa://cluster/myservice”目录下的文件，这样就可以获得 server1 和 server2，客户端随后可以从中选择一个节点发出自己的请求，从而实现负载均衡。

2．远程过程调用（夸父）

在分布式系统中，不同计算机之间只能通过消息交换的方式进行通信。显式的消息通信必须通过 Socket 接口编程，而远程过程调用（Remote Procedure Call，RPC[9]）可以隐藏显式的消息交换，使得程序员可以像调用本地函数一样来调用远程的服务。

夸父（Kuafu）是飞天平台内核中负责网络通信的模块，它提供了一个 RPC 的接口，简化编写基于网络的分布式应用。夸父的设计目标是提供高可用（7×24 小时）、大吞吐量（Gigabyte）、高效率、易用（简明 API、多种协议和编程接口）的 RPC 服务。

RPC 客户端（RPC Client）通过 URI 指定请求需要发送的 RPC 服务端（RPC Server）的地址，目前夸父支持两种协议形式。

- **TCP**：例如，tcp://fooserver01:9000
- **Nuwa**：例如，nuwa://nuwa01/FooServer

与用流（stream）传输的 TCP 通信相比，夸父通信是以消息（Message）为单位的，支持多种类型的消息对象，包括标准字符串 std::string 和基于 std::map 实现的若干 string 键值对。

夸父 RPC 同时支持异步（asynchronous）和同步（synchronous）的远程过程调用形式。

- **异步调用**：RPC 函数调用时不等接收到结果就会立即返回；用户必须通过显式调用接收函数取得请求结果。
- **同步调用**：RPC 函数调用时会等待，直到接收到结果才返回。在实现中，同步调用是通过封装异步调用来实现的。

在夸父的实现中，客户端程序通过 Unix Domain Socket 与本机上的一个夸父代理（Kuafu Proxy）连接，不同计算机之间的夸父代理会建立一个 TCP 连接。这样做的好处是可以更高效地使用网络带宽，系统可以支持上千台计算机之间的互联需求。此外，夸父利用女娲来实现负载均衡；对大块数据的传输做了优化；与 TCP 类似，夸父代理之间还实现了发送端和接收端的流控（Flow Control）机制。

3．安全管理（钟馗）

钟馗（Zhongkui）是飞天平台内核中负责安全管理的模块，它提供了以用户为单位的身份认证和授权，以及对集群数据资源和服务进行的访问控制。

- 用户的身份认证（Authentication）是基于密钥机制的。
- 用户对资源的访问控制是基于权能（Capability）机制进行授权（Authorization）的。

Capability 是用于访问控制的一种数据结构，它定义了对一个或多个指定的资源（如目录、文件、表等）所具有的访问权限。用户访问飞天系统的资源时必须持有 Capability，否则即视为非法。打一个比方，如果把 Capability 理解为地铁票，乘坐地铁（对地铁的一种访问方式）的时候必须要有 Capability，即地铁票。

密钥对是基于公开密钥方法的，包括一个私钥和相对应的公钥。在飞天平台系统中，密钥对用于数字签名服务，以保证 Capability 的不可伪造。换句话说，私钥用于产生数字签名（如签发 Capability），公钥用于验证数字签名的有效性（如验证签发过的 Capability 的有效性）。

考虑到网络通信时任何通信节点都是不可信的，所以即使是飞天自身模块内部之间的通信也同样是需要认证和授权的，而且验证的机制也完全一样。

2.2.2　分布式文件系统（盘古）

盘古（Pangu）是一个分布式文件系统，盘古系统的设计目标是将大量通用机器的存储资源聚合在一起，为用户提供**大规模**、**高可靠**、**高可用**、**高吞吐量**和**可扩展**的存储服务，是飞天平台内核中的一个重要组成部分。

- **大规模：**能够支持数十 PB 量级的存储大小（1PB=1000TB），总文件数量达到亿量级。
- **数据高可靠性：**保证数据和元数据（Metadata）是持久保存并能够正确访问的，保证所有数据存储在处于不同机架的多个节点上面（通常设置为 3）。即使集群中的部分节点出现硬件和软件故障，系统能够检测到故障并自动进行数据的备份和迁移，保证数据的安全存在。
- **服务高可用性：**保证用户能够不中断地访问数据，降低系统的不可服务时间。即使出现软硬件的故障、异常和系统升级等情况，服务仍可正常访问。
- **高吞吐量：**运行时系统 I/O 吞吐量能够随机器规模线性增长，保证响应时间。
- **高可扩展性：**保证系统的容量能够通过增加机器的方式得到自动扩展，下线机器存储的数据能够自动迁移到新加入的节点上。

同时，盘古系统也能很好地支持在线应用的低延时需求。在盘古系统中，文件系统的元数据存储在多个主服务器（Master）上，文件内容存储在大量的块服务器（Chunk Server）上。客户端程序在使用盘古系统时，首先从主服务器获取元数据信息（包括接下来与哪些块服务器交互），然后在块服务器上直接进行数据操作。由于元数据信息很小，大量的数据交互是客户端直接与块服务器进行的，因此盘古系统采用少量的主服务器来管理元数据，并使用 Paxos 协议[15]保证元数据的一致性。此外，块大小被设置为 64MB，进一步减少了元数据的大小，因此可以将元数据全部放到内存里，从而使得主服务器能够处理大量的并发请求。

块服务器负责存储大小为 64MB 的数据块。在向文件写入数据之前，客户端将建立到 3 个块服务器的连接，客户向主副本（Replica）写入数据以后，由主副本负责向其他副本发送数据。与直接由客户端向 3 个副本写入数据相比，这样可以减少客户端的网络带宽使用。块副本在放置的时候，为保证数据可用性和最大化地使用网络带宽，会将副本放置在不同机架上，并优先考虑磁盘利用率低的机器。当硬件故障或数据不可用造成数据块的副本数目不满 3 份时，数据块会被重新复制。为保证数据的完整性，每块数据在写入时会同时计算一个校验值，与数据同时写入磁盘。当读取数据块时，块服务器会再次计算校验值与之前存入的值是否相同，如果不同就说明数据出现了错误，需要从其他副本重新读取数据。

在线应用对盘古系统提出了与离线应用不同的挑战：OSS、OTS 要求低时延数据读写，ECS 在要求低时延的同时还需要具备随机写的能力。针对这些需求，盘古系统实现了事务日志文件和随机访问文件，用于支撑在线应用。其中，日志文件通过多种方法对时延进行了优化，包括设置更高的优先级、由客户端直接写多份拷贝而不是用传统的流水线方式、写入成功不经过 Master 确认等。随机访问文件则允许用户随机读写，同时也应用了类似日志文件的时延优化技术。

2.2.3 资源管理和任务调度（伏羲）

伏羲（Fuxi）是飞天平台内核中负责资源管理和任务调度的模块，同时也为应用开发提供了一套编程基础框架。伏羲同时支持强调响应速度的在线服务和强调处理数据吞吐量的离线任务。在伏羲中，这两类应用分别简称为 Service 和 Job。

在资源管理方面，伏羲主要负责调度和分配集群的存储、计算等资源给上层应用；管理运行在集群节点上任务的生命周期；在多用户运行环境中，支持计算额度、访问控制、作业优先级和资源抢占，在保证公平的前提下，达到有效地共享集群资源。

在任务调度方面，伏羲面向海量数据处理和大规模计算类型的复杂应用，提供了一个数据驱动的多级流水线并行计算框架，在表述能力上兼容 MapReduce[12]、Map-Reduce-Merge 等多种编程模式；自动检测故障和系统热点，重试失败任务，保证作业稳定可靠运行完成；具有高可扩展性，能够根据数据分布优化网络开销。

伏羲中应用了“Master/Worker”工作模型。其中，Master 负责进行资源申请和调度、为 Worker 创建工作计划（Plan）并监控 Worker 的生命周期，Worker 负责执行具体的工作计划并及时向 Master 汇报工作状态（Status）。此外，Master 支持多级模式，即一个 Master 可以隶属于另外一个 Master 之下。

伏羲 Master 负责整个集群资源管理和调度，处理 Job/Service 启动、停止、Failover 等生命周期的维护。同时伏羲 Master 支持多用户额度配置、Job/Service 的多优先级设置和动态资源抢占逻辑，可以说是飞天平台的“大脑”。伏羲对资源调度是多维度的，可以根据 CPU、内存等系统资源，以及应用自定义的虚拟资源对整个机群进行资源分配和调度。

土伯（Tubo）是部署在每台由伏羲管理的机器上的后台进程，负责收集并向伏羲 Master 报告本机的状态，包括系统资源的消耗、Master 或 Worker 进程的运行、等待、完成和失败事件，并根据伏羲 Master 或者 Job/Service Master 的指令，启动或杀死指定的 Master 或 Worker 进程。同时土伯还负责对计算机健康状况进行监控，对异常 Worker（比如内存超用）进行及时的清理和汇报。

对于在线服务（Service），由伏羲 Master 负责 Service Master 的启动与状态监控，处理相应 Service Master 的资源申请请求。Service Master 负责管理 Service Worker 的任务分配、生命周期管理以及 Failover 的管理。

对于离线任务（Job），伏羲 Master 负责 Job Master 的启动与状态监控，处理相应 Job Master 的资源申请请求。Job Master 根据用户输入的 Job 描述文件，将任务分解成一个或以上的 Task，每个 Task 的资源申请、Task Worker 的调度和生命周期维护由 Task Master 负责。

1. 在线服务调度

在飞天平台内核中，每个 Service 都有一个 Service Master 和多个不同角色（Role）的

Service Worker，它们一起协同工作来完成整个服务的功能。Service Master 是伏羲 Master 管理下的子 Master（Child Master），它负责这个 Service 相关的资源申请、状态维护以及故障恢复，并定期与伏羲 Master 进行交互，确保整个 Service 正确、正常地运行。每个 Service Worker 的角色和执行的动作，都是由用户来定义的。

每个 Service Worker 负责处理一个到多个数据分片（Partition），同一时刻一个分片只会被分配到一个 Service Worker 处理。将数据分割成为互不相关的分片，然后将不同分片给不同 Service Worker 来处理是构建大规模应用服务的关键特性。数据分片是一个抽象的概念，在不同的应用中有不同的含义。

在服务运行的过程中，每个 Service 的数据分片的数目和内容都是可以动态变化的，应用程序可以根据实际需要对数据分片动态地进行加载（Load）、卸载（Unload）、分裂（Split）和迁移（Migrate）等操作。

2．离线任务调度

在飞天平台中，一个离线任务（Job）的执行过程被抽象为一个有向无环图（Directed Acyclic Graph，DAG）：图上每个顶点对应一个 Task，每条边对应一个 Pipeline。一个连接两个 Task 的 Pipeline 表示前一个 Task 的输出是后一个 Task 的输入。

每个离线任务都有一个 Job Master 负责根据用户输入的任务描述（Job description）构造 DAG 和调度 DAG 中所有 Task 的执行。每个 Task 的 Task Master 会根据要处理的实例数量、数据在集群的分布及处理实例的资源需求，向伏羲 Master 申请机器资源并分配 Task Worker 在其上执行。分配到每台机器上的实例（Instance）是由 Task Worker 来具体执行完成的。每台机器上的 Task Worker 可以根据需要选择多线程或者多进程的不同运行模式。

在离线 Job 的容错方面，除了提供对异常机器的黑名单机制、长尾 Instance 的后备 Worker 机制外，伏羲还提供了快照（Snapshot）机制。快照是 Task 级别的容错机制。如果一个 Task 的 n 个 Instance 在前一次运行失败时完成了 m 个，那么 Task 重启后只会重新调度运行剩余的 $n-m$ 个 Instance。

2.2.4 集群监控和部署

1．集群监控（神农）

神农（Shennong）是飞天平台内核中负责信息收集、监控和诊断的模块。它通过在每台物理机器上部署轻量级的信息采集模块，获取各个机器的操作系统与应用软件运行状态，监控集群中的故障，并通过分析引擎对整个飞天的运行状态进行评估。

神农系统包括 Master、Inspector 和 Agent 三个部分。

- **Master：**负责管理所有神农 Agent，并对外提供统一的接口来处理神农用户的订阅（Subscription）请求，在集群中只有一个 Master。

- **Inspector**：是部署在每一台机器上的进程，负责采集当前机器和进程的通用信息，并实时发送给该机器上的神农 Agent。
- **Agent**：是部署在每台物理机器的后台程序。Agent 负责接受来自应用和 Inspector 写入的信息。Agent 启动后，会立刻向 Master 注册自己，并根据 Master 发来的订阅（Subscription）命令执行相应的信息采集、过滤、聚合和处理操作。目前神农 Agent 处理的数据分为两类：事件类数据（如应用程序故障和报警）和数值类数据（如当前应用的性能计数、机器 I/O 吞吐量等）。

神农的用户通过 Master 来访问神农系统，以数据订阅（Subscription）的方式获取神农系统采集到的信息。

神农的 MonitorService 和 AnalysisService 是使用神农系统的两个应用程序。

- MonitorService 在集群中的一台机器上部署，通过向各个 Agent 发送特定的监控请求，并根据配置设定的规则，实现对集群的状态和事件的监控，以及报警和记录。
- AnalysisService 也是部署在集群中的一台机器上，通过访问神农来获得主要性能数据，然后聚合数据并计算出系统的总体资源情况（例如，集群的总资源消耗、总 I/O 吞吐量等），并且向外提供计算结果供查询。

2．集群部署（大禹）

大禹（Dayu）是飞天内核中负责提供配置管理和部署的模块，它包括一套为集群的运维人员提供的完整工具集，功能涵盖了集群配置信息的集中管理、集群的自动化部署、集群的在线升级、集群扩容、集群缩容，以及为其他模块提供集群基本信息等。每个飞天模块的发布包都包含一个部署升级的描述文件，定义了该模块部署和升级的流程，提供给大禹使用。

在结构上，大禹包含了集群配置数据库、节点守护进程、客户端工具集等部分。

集群配置数据库负责存放和管理所有部署了飞天的集群的配置信息，包括集群中每个节点承担的角色、各个模块的软件版本、各个模块的基本参数配置等。同时，数据库中还记录了部署或升级时每个节点的任务执行状态，保证了在部署或升级时少量不在线节点可以在重新连线后进行自动修复。

节点守护进程运行在集群的每一个节点上，负责与集群配置数据库同步该节点相关的集群信息，执行节点相关的具体运维任务，并汇报任务执行状态。节点守护进程本身是自我升级的，只需部署一次，即能保证运行的是该集群最适合的版本。在模块软件部署和升级的过程中，节点守护进程还负责软件的下载分发，为了保证效率和规避单点故障，软件的分发采用 P2P 的方式进行。

客户端工具集是运维人员实际使用的命令行工具和网页界面，运维人员通过这些工具对集群进行部署、升级、扩容、缩容等具体操作。大部分操作都提供了自动化和人机交互执行两种方式，分别适应简便操作和精细化控制这两种场景。在部署和升级的过程中，客

户端工具负责控制总体的操作顺序，维护模块之间的依赖关系，并根据状态信息决定是否回滚或中断当前流程。

2.3 飞天开放服务

本节从整体上简要介绍飞天开放服务，包括弹性计算服务（ECS）、开放存储服务（OSS）、开放结构化数据服务（OTS）、关系型数据库服务（RDS）、开放数据处理服务（ODPS）和云服务引擎（ACE）。这些开放服务运行在飞天平台内核之上，具有以下一些共同的特点。

- **全托管式服务：**开放服务运行在数据中心的公共云平台之上，用户无须关心硬件设备的采购和软件系统的配置、管理，这些服务以全托管的方式为用户提供直接可用的软件服务。这样，用户可以专注在应用层逻辑的设计与实现，按照实际使用的多少进行付费，因此减少了初期在基础设施上的投入，节省了应用的成本。此外，开放服务还向用户提供详细的资源使用统计、性能指标和操作日志，方便用户调查错误和分析应用的行为。开放服务由阿里云的专业人士进行维护和优化，提供高端的基础设施和网络安全保障，用户无须担心数据备份、故障恢复和扩展升级等方面的问题。
- **数据安全可靠：**开放服务都采用盘古作为底层的存储，所有数据都为多份冗余存储。底层存储系统会自动处理集群中的硬件和软件错误，对用户屏蔽这些错误。此外，用户的数据在存储层完全被隔离，用户对数据的访问必须通过身份验证的机制，有效地保障了用户数据的安全和隐私。
- **可扩展性：**开放服务提供的资源完全可以随着用户使用负载的变化而弹性伸缩，用户只需要专注自身最核心的业务，而不用担心数据量的激增带来的数据可靠性和客户访问的性能问题。例如，在 OTS 服务中，系统通过对表进行横向切分（Partitioning）来实现规模的扩展，数据均匀地散落到多个存储节点上，可以通过增加机器和调整调度实现服务整体规模的扩展。

2.3.1 弹性计算服务（ECS）

弹性计算服务（ECS）为用户提供一个根据需求动态运行的虚拟服务器的环境。对于 ECS 提供的虚拟服务器，用户可以像使用一台物理机器一样进行各种操作。ECS 允许用户根据自己的需要，租用多台虚拟服务器来完成各种任务。在运行的过程中，用户也可以根据计算资源的需要动态增加或减少虚拟服务器的数量。

对于用户来说，弹性计算服务解决了业务的周期性变化带来的资源利用率不高和 IT 成本高的问题。同时，弹性计算服务还可以减少 IT 采购的周期，提供数据的可靠存储和

可扩展的能力，并可以有效地减少网络安全的威胁。

针对弹性计算服务，阿里云还提供了云监控、云盾和负载均衡这三个产品。

- 云监控为云服务器提供第三方监控服务，可以及时发现故障并通过多种方式报警，包括网站、Ping、TCP 端口、UDP 端口、DNS、POP3、SMTP、FTP 等监控。云监控除了可以为 ECS 提供安全有效的监控服务外，还能为其他自由服务器提供监控服务，用户只需要通过简单的配置即可实现各种监控需求。
- 云盾为云服务器提供一站式安全增值服务，包括安全体检（网页漏洞检测、网页挂马检测）、安全管家（防 DDOS 服务、端口安全检测、网站后门检测、异地登录提醒、主机密码暴力破解防御）等功能。
- 负载均衡（Server Load Balancer，SLB）通过设置虚拟 IP，将位于同一数据中心的多台云服务器资源虚拟成一个高性能、高可用的应用服务池，再根据应用特性，将来自客户端的网络请求分发到云服务器池中。SLB 会检查池中云服务器的健康状态，自动隔离异常状态云服务器。同时，SLB 还可以增强云服务器池的抗攻击能力、安全隔离应用和云服务器。云服务器无须特殊设置即可透明接入 SLB。

2.3.2 开放存储服务（OSS）

开放存储服务（OSS）是阿里云对外提供的海量、安全、低成本和高可靠的云存储服务。OSS 支持海量的文件存储，同时在多个地方调用呈现，极大地简化了用户数据管理、迁移和更新的工作。用户可以通过简单的 RESTful API（RESTful API 的介绍参见附录 B.1），在任何时间、任何地点、任何互联网设备上进行上传和下载数据，也可以使用 Web 页面对数据进行管理。OSS 目前已经在多个云存储服务、电子商务网站和手机应用网站中使用，提供包括图片、软件和音视频文件在内的存储和互联网访问服务。

在 OSS 中，用户文件都是以 Object 的方式存储，每个 Object 包含名称、数据和用户存储的关于 Object 的元数据（Metadata）。由于 OSS 中 Object 不允许重命名和部分修改，因此，OSS 服务适合于存储写一次、读多次的数据，例如，视频、音频、图片和备份文件等。OSS 支持对整个 Object 内容进行替换的修改操作。

OSS 的命名空间采用 Bucket 的方式：每个 Bucket 中可以存储任意数量的 Object，但 Bucket 本身并不直接包含任何数据。存储在 OSS 上的每个 Object 必须都属于某个 Bucket，Bucket 名在整个 OSS 系统中具有全局唯一性，且不能修改。如果一个 Bucket 名已经被某个用户使用，那么其他用户都不能再使用这个 Bucket 名。OSS 目前提供 Bucket 级别的访问权限控制，包括 public-read-write、public-read 和 private 这三种访问权限。

2.3.3 开放结构化数据服务（OTS）

开放结构化数据服务（OTS）是阿里云对外提供的支持海量结构化和半结构化数据存

储与实时访问的服务。OTS 以表的方式存储数据，保证强一致性。一个用户可以拥有多个表，每个表中包含任意多行数据，每一行又可以包含任意多个列，除主键外的列不需要在创建表时指定。OTS 还支持视图、表组和事务等高级功能。用户可以在表中查询、插入、修改和删除数据。用户可以通过 RESTful API 来使用服务，也可使用 Web Portal 页面对数据进行管理。

OTS 目前在多个互联网应用场景中得到成功的使用，提供结构化数据的存储和实时访问服务。用户使用 OTS 可以免去雇用专人来管理和维护数据库软件的开销。OTS 服务按实际使用量付费的方式也降低了客户的使用成本。用户也无须担心随着应用规模的不断扩大，数据量和并发访问的可扩展性需求，OTS 服务通过自动扩展的方式为应用的长期快速发展解决后顾之忧。

2.3.4 关系型数据库服务（RDS）

关系型数据库服务（RDS）通过 Web 方式为用户提供可以在几分钟内生成并投入生产的、经过优化的数据库实例，支持 MySQL 和微软 SQL Server 这两种关系型数据库，适合于各行业中小企业的关系型数据库应用。使用阿里云的 RDS 服务能够使得中小企业根据业务规模发展的需要快速部署适合自己的数据库实例，因而无须购买昂贵的硬件和聘用管理维护人员，降低了企业使用数据库的综合成本。

RDS 提供的数据库与用户自己搭建的数据库环境和使用方式完全相同，用户只需要使用通用的数据导入导出工具即可直接将已有的数据库迁移至 RDS 服务中。由于 RDS 数据库硬件和数据都部署在云端，利用阿里云提供的基础设施、网络安全保障、专业的系统运维维护及热备服务，数据库的备份、恢复和扩展升级等日常管理功能都极大地得到了简化。

以上 RDS 提供的各项功能及服务都不需要前期投资，用户只需要根据使用量进行付费即可。传统企业自建数据库的方式一般存在设备利用率偏低、不能按需部署、无法快速应对规模变化以及投入成本过高、维护成本高和建设周期过长等问题。而 RDS 相对于用户自建数据库具有低成本、高效率、高可靠、灵活易用等优点，使企业有更多的时间聚焦于自身的核心业务上面。

2.3.5 开放数据处理服务（ODPS）

开放数据处理服务（ODPS）提供了大规模数据的离线处理和分析服务，它以 RESTful API 的形式支持基于描述性查询语言 SQL 的数据处理，并提供 MapReduce[12]的并行计算框架。ODPS 重点面向数据量大（PB 级别）且实时性要求不高的海量数据分析应用，适用于海量数据统计、数据建模、数据挖掘、数据商业智能等互联网应用。

ODPS 提供了 SQL 与 MapReduce 两种 API 供用户开发调用。ODPS SQL 采用类似 SQL 的语法来处理大规模（PB 级别）数据，适合于处理强调数据吞吐量的离线任务。ODPS SQL

提供了大量操作海量数据的 SQL 语法支持（API），例如，创建、删除表和视图的 DDL 语法，更新表的 DML 语法等。为了方便用户完成数据处理的各类任务，ODPS SQL 还提供了很多高级功能，例如，窗口函数、用户自定义函数、存储过程等。与数据库相比，ODPS SQL 并不具备数据库的一些特征，包括事务和主键约束。ODPS SQL 的优势在于能够快速处理海量数据，它能够将多个 SQL 语句以它们之间的数据依赖关系组成一个工作流，然后以执行工作流的方式完成复杂的数据分析功能。

ODPS 的 MapReduce 语法与 Hadoop[19] MapReduce 类似，基于此编程框架编写的程序以一种可靠容错的模式运行在由数千个通用服务器搭建的大规模集群上，能并行处理 PB 级别的海量数据。与 Hadoop 上使用的 MapReduce 相比，ODPS 为用户提供了开箱即用（Out-of-Box）的离线数据处理环境，用户在注册 ODPS 账号以后即可使用。这样，用户可以集中精力于业务逻辑的实现上，而不用关心环境的搭建、配置、监控和调优。

2.3.6 云服务引擎（ACE）

云服务引擎（ACE）是飞天平台提供的一个基于云计算基础架构的网络应用程序托管环境，帮助应用开发者简化网络应用程序的构建和维护，并可根据应用访问量和数据存储的增长进行动态扩展。

ACE 支持 PHP 和 Node.js 语言编写的应用程序，支持标准的关系型数据库（例如 MySQL）、Memcache、Cron、Session 和 Storage，同时增加一些高级特性来满足开发者的需求。ACE 选择 PHP 作为首选支持语言，ACE 的 PHP Runtime 和官方标准 PHP 环境几乎完全一样，99%的代码可以不加任何修改就可以完美地运行在 ACE 环境中。出于安全和性能的考虑，ACE 对标准 PHP 进行了一些扩展和改进。

截至本书出版时，ACE 还在开发中，相应的 API 也没有对外开放。因此，在本书接下来的章节中不单独详细描述 ACE，只是在附录 D 中介绍移动终端云应用开发时，简单介绍云应用怎样利用 ACE 空间来开发云端服务接口。