

| 第二部分 |

Mastering Cloud Computing: Foundations and Applications Programming

# 云应用编程与 Aneka 平台

## Aneka: 云应用平台

Aneka 平台是 Manjrasoft 公司推出的关于云应用开发、使用、管理的解决方案。Aneka 包括一个可在异构计算平台上部署的可扩展云中间件。它为云应用提供了一系列辅助功能来协调应用程序的执行，帮助管理者监视云中的状态，同时也提供现有云技术的整合。Aneka 的核心优势之一是针对不同编程模型（例如任务、线程、MapReduce）提供了可扩展应用编程接口（API），用于不断发展分布式应用并将新的技术整合到云应用中，同时 Aneka 还支持不同的云部署方案（公共云、私有云、混合云，如图 5-1 所示）。这些特性使 Aneka 不同于 IaaS 平台上的管理软件，成为一个可在多种云平台上进行云应用开发、部署和管理的综合平台。

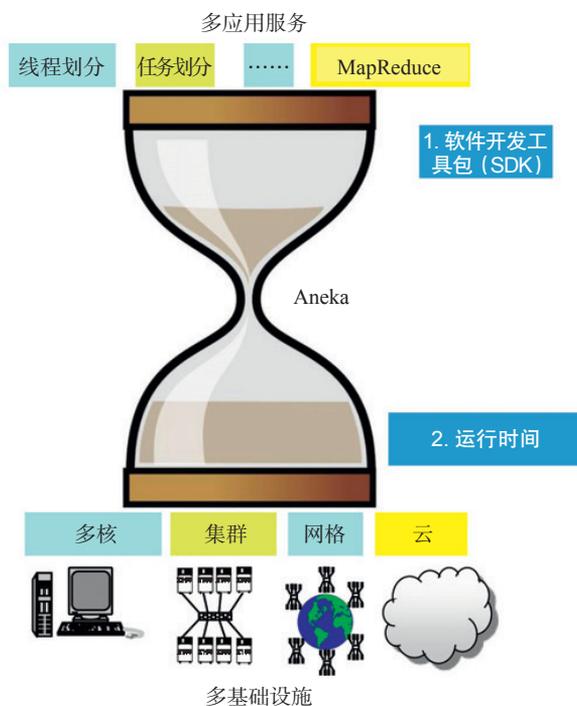


图 5-1 Aneka 功能一览

本章对 Aneka 系统框架进行了描述和总结，介绍了构成 Aneka 云平台的组件和基本服务，同时讨论了一些具有普遍性的云应用部署场景。

### 5.1 框架概述

Aneka 是一个开发云应用的软件平台，它可以利用分布式系统资源在一个独特的运行云应用的虚拟域——Aneka 云平台中对其进行管理。根据第 1 章所提供的云计算参考模型，

Aneka 是一个纯平台即服务的云计算解决方案。Aneka 是一个可以部署在异构资源平台上的云中间件产品，例如局域网中的计算机、多核服务器、数据中心、虚拟化的云设备或上述几种的混合。Aneka 框架不仅提供了管理和扩展分布式云应用的中间件，同时也提供了开发分布式云应用的编程接口集合。

图 5-2 对 Aneka 框架的组件进行了全面的概括性展示，系统的核心框架提供了统一的接口层，使 Aneka 框架可以部署在不同的平台和操作系统上。物理和虚拟的资源池代表了云中有限的资源，这些资源由 Aneka 容器（一种轻量级虚拟化技术）管理。Aneka 容器安装在每个节点上，构成了云中间件的基本单元。一系列互联的容器构成了 Aneka 云平台：一个向用户和开发者开放服务形式的统一域。容器提供了三种不同特性的服务形式：构造服务、基础服务和执行服务。这些服务分别负责设备管理、云支撑服务、云应用管理和运行。这些服务通过应用管理和开发接口层提供给开发者和管理人员，包括云应用开发 API 和云平台控制管理工具。

141  
143

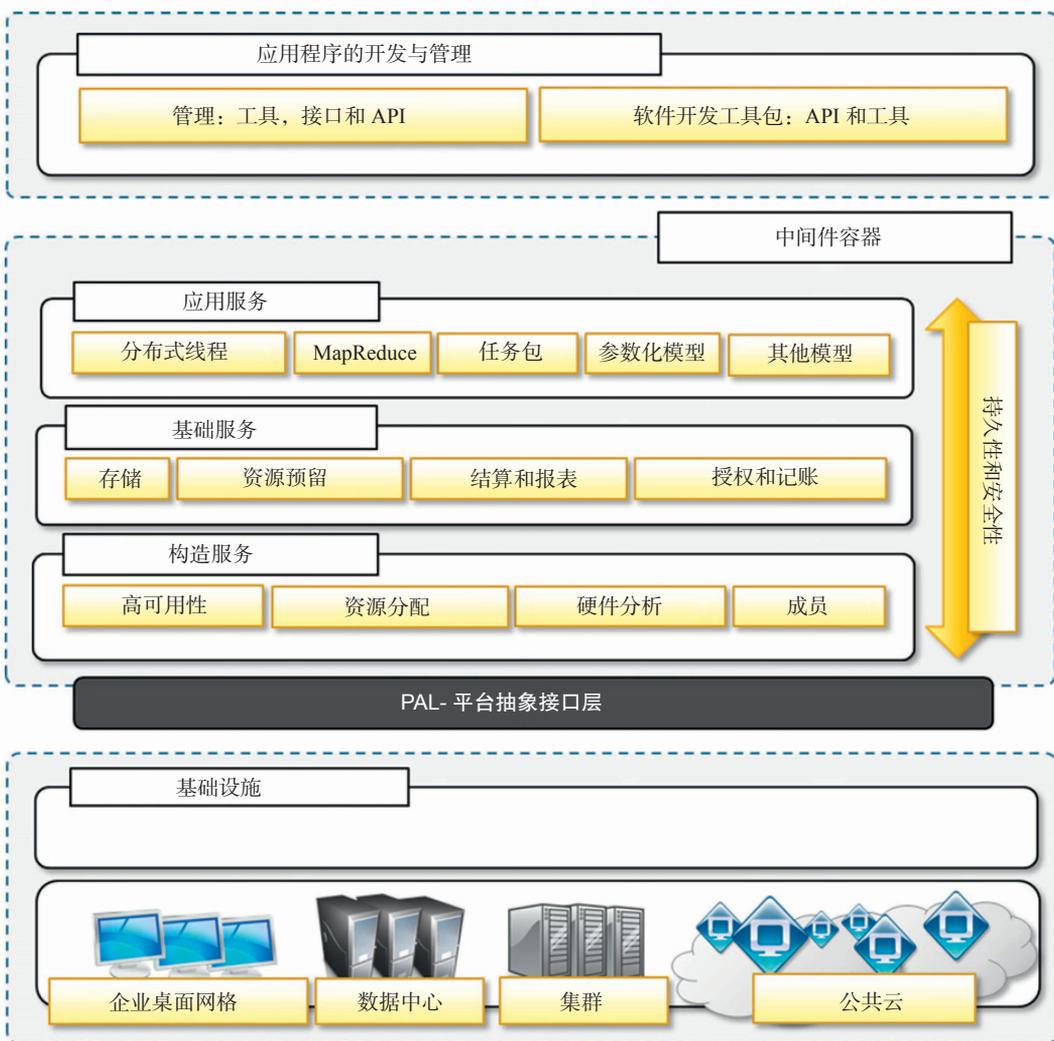


图 5-2 Aneka 框架概述

Aneka 继承了面向服务架构 (SOA) 的全部特性, 服务是 Aneka 云平台的基本组件。服务接口建立在容器层上, 框架中平台抽象层的特性对于用户、开发者和管理者是不可见的。Aneka 云平台同时提供了扩展和重新定义服务实现的服务接口, 允许多种新服务的整合以及用不同的实现替代现有实现。该框架包括基础设施、节点管理、应用程序执行、计费 and 系统监控等基本服务, 现有的服务可以进行扩展, 同时新的功能可以通过动态插入容器的方式添加进云平台。容器虚拟化技术的可扩展性和灵活部署使 Aneka 云平台能够有效支持不同应用的编程模型和执行模型。这里的编程模型是指一个抽象集合, 程序员可以用它来表述分布式应用。编程模型的运行支持由一个集合构成, 该集合通过执行与服务相互作用来推动应用的执行。因此, 一个新模型的实现需要由应用开发人员和 service 所使用的特定编程抽象的发展来为其提供运行支持。编程模型只是云应用管理和运行的一个方面, Aneka 云平台还提供云应用的弹性资源分配和分布式协作服务。这些服务包含以下内容:

[144]

- 弹性和可扩展性。通过动态分配服务, Aneka 支持对应用提供资源容量的增减。
- 运行管理。运行机制需要维护基础设施的可用性以支撑应用运行, Aneka 平台主要由容器技术、服务成员管理、基础设施监控和维护、程序概要分析等服务接口实现运行管理。
- 资源管理。根据应用需求和用户定制, Aneka 可以弹性提供设备资源。为了提供基于服务质量 (QoS) 的运行环境, 系统不仅允许动态资源备份也允许为某些应用保留独占节点。
- 应用管理。应用管理对应一个专属的服务子集, 子集中的服务包括调度、执行、监控和存储管理。
- 用户管理。Aneka 是一个多租户的分布式环境, 提供一个扩展的用户系统来定义用户、群组和权限许可。用户管理下的服务接口构成了系统的安全基础机制, 以及计算管理的基础服务组件。
- QoS/SLA 管理和结算。云环境将会对执行的应用进行计量和资费结算。Aneka 提供一系列协同服务来监控每个应用的资源使用情况并向用户提供账单。

[145]

所有上述服务由软件开发工具箱 (SDK) 和管理工具箱提供的应用程序接口 (API) 实现。SDK 的 API 向开发者提供现有编程模型和对象模型来创建新的模型。管理工具箱集成了运行服务接口来管理基础设施、用户和应用。管理工具箱对云平台进行整体状态描述和监控, SDK 则更多地提供对于单个应用执行的服务接口。两组工具箱致力于为核心虚拟化容器提供交互和管理的易用接口。

## 5.2 Aneka 容器结构

Aneka 容器是 Aneka 云平台的基础组件, 代表了服务及应用的运行机制。容器是 Aneka 的资源部署的基本单位, 作为轻量级的软件中间件来承载服务功能, 与接入的操作系统和硬件进行交互。容器提供了一个轻量级的服务部署环境, 可用于部署服务和一些基础功能, 比如和云中其他节点通信的信道接口。云平台的所有操作都由容器所承载的服务接口提供。容器上加载的服务可以分为三个基本类型:

- 构造服务。
- 基础服务。
- 应用服务。

上述服务构成的服务栈位于平台抽象层 (PAL) 之上, 提供了对潜在操作系统和硬件的接口, 从容器的角度提供了软硬件资源。持久性和安全性横跨了所有服务栈的内容, 提供了一个安全可靠的框架结构。下面章节将具体讨论这些服务层次的细节。

146

### 5.2.1 Aneka 平台基础: 平台抽象层

系统的核心架构基于 .NET 技术, 使 Aneka 能够在不同平台和操作系统上迁移。任何满足 ECMA-334[52] 和 ECMA-335[53] 规范的平台环境均可在 Aneka 容器上加载和运行。

ECMA-334 标准引入了通用语言基础设施 (CLI), 定义了一个通用代码运行环境, CLI 不会向运行在其框架上的应用程序提供任何接口来访问硬件或收集宿主操作系统的运行信息<sup>⊖</sup>。此外, 不同的操作系统有不同的文件系统结构和存储方式。平台抽象层 (PAL) 可以屏蔽这种异构性, 向容器提供统一接口来访问硬件和操作系统信息, 因此 Aneka 容器可以在任何 PAL 所支持的平台运行。

PAL 负责检测宿主环境的兼容性并实现与宿主环境的交互, 从而支持上层容器的运行。PAL 具有如下特征:

- 提供统一且与平台无关的执行接口来访问宿主平台。
- 对宿主平台的扩展和附加属性提供统一访问途径。
- 提供统一且与平台无关的途径来访问远端节点。
- 提供统一且与平台无关的管理接口。

PAL 是 Aneka 中一个小的接口层构成的检测引擎, 启动后会自动对容器进行配置。PAL 实现了具体硬件资源的接入和对不同操作系统 (Windows、Linux 和 Mac OS X) 的抽象层构建。

PAL 向上层提供的信息包括:

- 内核数量、频率和 CPU 用途。
- 内存的大小和使用情况。
- 全局层面的硬盘可用空间。
- 网络地址和节点上的设备。

此外, 通过查询硬件属性可以获得附加硬件配置信息。PAL 提供定制实现的方法, 通过使用表述平台信息的命名 - 值来提取附加信息。例如, 属性可以包含 CPU 的型号和系列信息, 以及容器所在的进程信息。

### 5.2.2 构造服务

构造服务定义了 Aneka 容器最底层的软件栈。构造服务提供了资源分配子系统和设备监控系统的接口。资源提供服务负责利用虚拟化技术按需求动态分配节点。监控服务可以对硬件信息进行分析, 同时, 作为一个基础监控设施, 安装在容器中的任何应用都可以使用这一服务。

147

#### 1. 概要分析和监控

概要分析和监控服务主要由心跳、监控和报告服务构成。心跳服务从 PAL 收集信息, 监控和报告服务对云中任意服务行为实行一般性监控。

⊖ .NET 虚拟机 CLR 是 CLI 的一种实现。——译者注

心跳服务周期性地从节点收集动态运行信息并将这些信息发布给云中的成员服务。这些信息由云的索引节点收集，为资源预留和调度服务提供必要依据，实现异构基础设施的优化应用。如前文所述，除了收集内存、硬盘剩余信息、CPU 和操作系统等基础信息以外，一些附加信息（例如系统中安装的软件）也可以加入到动态报文信息中。更确切地说，任意以文本描述的属性信息都可以在报文中传递。一个名为 `node resolver` 的服务组件负责收集上述运行信息并传递给心跳服务。Aneka 为该服务组件提供了多种方法以满足不同异构环境下的实现。PAL 使用不同接口来对接不同操作系统，部署在不同节点上的 Aneka 组件可以屏蔽操作系统的异构性，以获取类型一致的信息。例如，在物理机和 IaaS 服务商（EC2、GoGrid）提供的虚拟机上获取公共 IP 的机制是不同的。Aneka 在虚拟容器部署中为每个节点配置一个 `resolver` 服务组件，以保证系统其他组件工作的透明性。

系统集成的监控和概要分析服务通过一个通用监控平台实现，该平台允许任何用户服务报告其行为。该平台由报告和监控服务接口构成，报告服务负责监控数据的存储并将监控数据提供给其他服务接口或者分析应用。每个节点上会实例化一个监控服务作为报告服务的网关，向报告服务传送采集节点上的监控数据。任何想要提供监控信息的服务可以借助本地的监控服务接口而不必了解整个基础设施的细节。目前已经有以下集成服务通过这种途径来提供相关信息：

- 成员目录记录节点运行信息。
- 执行服务在任务执行过程中选取时段进行监控。
- 调度服务记录任务的状态信息。
- 存储服务监控和提供数据迁移中的信息，例如上传和下载次数、文件名和大小。
- 资源分配服务记录虚拟节点的分配和生命周期信息。

上述信息可以存储在一个关系型数据库管理系统（RDBMS）或展平文件<sup>⊖</sup>中，以便应用进行深入分析。例如，管理控制台提供了管理层面的数据监控信息。

[148]

## 2. 资源管理

资源管理是 Aneka 云平台的另一项基本特征，它需要完成以下几个任务：资源组成、资源预留和资源分配。Aneka 提供了一系列服务接口负责资源管理，包括索引服务（即成员目录）、预留服务和资源分配服务。

成员目录是 Aneka 资源管理的基本组件，它记录了所有节点（无论连接与否）的基本信息。成员目录构成了目录服务的一个基础服务，允许通过节点名等属性字段来搜索服务。当容器启动时，每个容器实例都会将自身信息发布到成员目录并在生命周期内不断更新。服务和外部应用可以查询成员目录以发现可用容器并与其进行交互。为了加速查询，成员目录设计成一个分布式数据库：查询先在本地记录的缓存中进行，如果没有结果再向主索引节点查询，主索引节点包含全局的成员目录。成员目录同时保存每个节点的动态信息，这些信息被发送到本地监控服务实例中并被长期保存。

资源的索引和划分是资源管理的基本功能，这些功能是在基础索引服务的接口上实现的。容器实例的部署和配置在设备管理层实现，并不属于构造服务。

动态资源分配允许将从 IaaS 服务商处租用的虚拟资源整合到 Aneka 云平台中并进行管理，该服务改变了 Aneka 云平台中的拓扑结构，允许其根据不同需求（例如处理节点故障、

⊖ 数据库的无结构化的普通文件存储形式。——译者注

保证应用的服务质量、维持云的稳定性能或恒定吞吐量)来进行扩展部署。Aneka 定义了一个非常灵活的框架来实现资源分配,其中资源动态申请的触发机制、添加容器数量的选择以及添加容器后的新运行策略都是可以修改的。Aneka 集成的资源分配平台主要集中在资源分配服务,该服务提供了分配虚拟容器实例资源的所有操作。资源分配服务的实现是基于资源池的设计理念,资源池提供了一个通用抽象接口来实现与具体 IaaS 服务商资源的交互,使所有服务商的资源池可以被统一管理。资源池并不一定要映射到 IaaS 服务商,也可以将由 Xen Hypervisor 管理的私有云或者只是偶尔使用的物理机资源作为动态资源。系统基于开放协议,允许使用元数据来描述资源池和定制资源分配请求。该平台简化了附加特征的实现,以及对能够被集成到现有系统的不同实现方法的支持。

资源分配功能是为了支持面向 Qos 服务需求的应用执行。因此,资源分配服务多数是从预留服务和调度服务发送的请求。此外,外部应用可以直接利用 Aneka 的资源分配功能,动态检索客户端,并与相关基础设施进行交互。这使得资源分配功能不仅可被 Aneka 使用,也可以被虚拟机管理器使用。

149

### 5.2.3 基础服务

构造服务是 Aneka 云平台的基础服务,定义了系统管理平台的基础功能。基础服务主要用于对构建在基础设施上的分布式系统进行逻辑管理,并为分布式应用的执行提供支持。所有支持的编程模型都可以集成进服务接口,同时利用这些服务接口提供高级且全面的应用管理。这些服务包括:

- 应用的存储管理。
- 资源的定价、支付和结算。
- 资源预留。

基础服务提供统一途径来管理分布式应用,使开发者只需关心编程模型的逻辑差异。结合构造服务和基础服务构成了 Aneka 中间件的核心。这些服务接口主要由执行服务和管理平台所使用。外部应用也可使用这些开放接口来提供高级应用管理。

#### 1. 存储管理

数据管理是分布式系统和云环境中的重要方面。应用对数据的操作多数以文件存储和移动的形式进行。因此,任何支持分布式应用的平台应该提供数据/文件迁移管理和持久化存储服务。Aneka 提供了两组存储管理的服务模式:集中式文件存储,主要适用于计算密集型应用;分布式文件系统,适合于数据密集型应用。两种应用的需求截然不同。计算密集型服务主要是需要高性能处理器,对存储没有太高要求,在许多实用案例中仅需要存储小型文件,文件在节点间的迁移也十分方便。在这种情况下,设置一个中心存储节点或者存储节点集群是较为合适的解决方案。相反,数据密集型应用的特点是处理大型数据文件(GB 或 TB),任务对处理能力的需求并不构成性能瓶颈。在这种情景下,利用分布式系统来管理所有云节点的存储空间是一个更好且更具可扩展性的解决方案。

集中式存储通过 Aneka 的存储管理服务接口来实现,该服务构成了 Aneka 的数据分级服务。存储服务向分布式应用提供了基本的文件传输功能以及面向终端用户或其他系统组件的抽象协议接口,接口可由安装到云中的服务实例动态配置(工厂模式)。目前默认的协议配置是 FTP。

为了支持不同的协议,系统引入文件通道的概念并定义了两个组件:文件通道控制器和

[150]

文件通道句柄。文件通道控制器构成了通道的服务组件，包含文件存储和访问权限。文件通道句柄代表了顾客组件，用户应用和系统其他组件可以调用句柄实现上传、下载和浏览文件。存储服务利用工厂模式下的文件通道首先创建管理存储的服务组件，然后按需求创建用户组件。需要文件传输支持的用户应用会自动分配文件通道句柄，实现输入文件的上传和外部文件的下载。同时，管理平台将会对工作节点进行配置，使节点能够完成执行文件的取回和结果上传。

文件通道抽象的一个令人感兴趣的特征是，链接两个文件通道以实现通过两种不同的文件协议来传输文件。Aneka 的每个文件中都包含元数据，可帮助管理平台选择合适的通道来移动文件。例如，一个输出文件可以利用内部 FTP 协议从工作节点传递给服务实例，再通过服务实例管理的 FTP 通道控制器输出到 AWS S3 服务的桶下。存储服务接口支持基于任务划分和线程模型的编程以及基于参数化的应用。

数据密集型的应用采用分布式文件系统进行存储，其参考模型是谷歌文件系统 (GFS)[54]，具有基于商用硬件的高度可扩展的基础设施。文件系统架构基于主节点，主节点中包含用于跟踪所有存储节点状态的文件系统全局映射，以及用于提供分布式存储空间的数据块服务器资源池。文件逻辑上存放在一个目录结构中，但同时通过一个唯一的 ID 产生分结构化域名，并以此作为索引存放在文件系统中。每个文件由一系列相同大小的块构成，每个文件块分配唯一的 ID 且存放在不同的服务器上，并进行副本存储来提供高可用性和容错性。该模型由 GFS 提出并能够对具有以下特征的应用提供优化支持：

- GB 级别以上的文件。
- 修改文件时只需要增加文件内容而不需要对原有文件内容进行修改。
- 以大量读取访问和少量随机访问为主的负载类型。
- 持续稳定的带宽比低响应延迟更为重要。

此外，考虑到文件系统中有大量的商用机器共同工作，因此进程故障和硬件故障是一种常态化现象。这些现象影响了存储机制的设计，使系统能够为应用的数据操作提供更好的性能。目前，只有 MapReduce[55] 能够使用分布式文件操作系统，MapReduce 也是 GFS 的主要应用。Aneka 在 Windows 文件系统基础上提供了一个简单的分布式文件系统。

## 2. 记账、结算和资源定价

记账服务记录了云中应用的状态，收集的信息对分布式节点的使用情况做出了细致划分，对资源管理的优化至关重要。

[151]

记账服务所收集的信息主要与设备使用和应用执行有关。应用执行过程中的完整历史信息以及存储等资源的使用参数将会被记录服务抓取和分析。这些信息构成了用户计费的基础信息。

结算是记账服务的另一个重要功能。Aneka 是一个多用户云编程平台，云中应用的执行可能直接涉及从 IaaS 服务商处调度更多资源，并产生协作成本。Aneka 结算服务为每个用户提供资源使用及其相关费用的详细信息。每项资源可以根据服务接口的类型或者节点上安装的软件来定价。结算模型提供了对各项应用的综合结算，对于具体用户提供每项任务的结算信息。

记账和结算功能由记账服务和报告服务提供。前者记录了应用的执行信息，例如任务在可用资源上的分配，各项任务时序及协作成本。后者提供了从监控服务接口获取的用于结算的信息，如存储使用情况和 CPU 性能。这些信息主要由管理控制台提取。

### 3. 资源预留

资源预留支持分布式应用的执行，使某些应用可以独占预留资源。资源预留由两种服务接口实现：资源预留服务和部署服务。资源预留服务记录了所有的预留时隙并提供系统的全局视角。部署服务安装在每个执行服务的节点上，管理本地节点的时隙分配。有截止时间的应用可以在一个时间段内通过资源预留申请一系列节点资源。如果预留请求可以满足，预留服务实例将会返回一个标识符作为资源预定凭据。应用执行期间，该标识符将会被用来选择执行应用的预留节点。在每个预留节点上，执行服务使用分配服务接口通过检验分配标识符来判断每项任务在执行时隙内是否具有占用许可权。尽管这是一个资源预留框架下的通用参考模型，但 Aneka 允许不同的服务实现方式，包括利用多种资源预留协议，以及在预留请求中配置相关参数。不同的协议和策略可以透明化地进行集成，Aneka 提供扩展 API 来支持高级服务。目前，系统框架支持三种不同的实现：

- 基本预留。代表节点时隙预留的基本方法，执行双备份协议，提供预留资源的备份选项以避免初始的预留请求不能被满足。
- 秤量（计价）预留。根据节点的硬件资源进行节点定价并提供预留。
- 中继预留。实现简单，可以使资源代理商在 Aneka 中预留节点并控制节点资源，该接口在互联云环境下的集成交互中比较有用。

152

资源预留是保证应用服务质量（预先经过协商）的基本功能，它可以保证云平台提供一个可预见的环境来控制应用的执行。预留请求允许与否的判断机制是基于请求时刻的可用资源静态分布，还要考虑目前和未来的工作负载。该方案对节点故障十分敏感，可能会使云平台无法满足服务等级协议 SLA。实际的服务应用机制倾向于尽可能推迟为预留请求分配节点，以应对临时故障和局部断电。但是对于大范围断电导致剩余节点不足以负载需求的情况，该策略将无法应对。为了应对这种情况，资源调度服务提供了一种更有效的解决途径：可以由外部资源服务商提供调度节点来应对断电和满足应用的 SLA 需求。当系统中的现有资源不足以满足预留请求时，现有的资源预留机制会使用构造层的调度方法来满足预留需求，解决资源不足和临时故障的问题。

#### 5.2.4 应用服务

应用服务对应用的执行进行管理，并针对不同的分布式负载应用对应的编程模型提供相应接口。由于每个编程模型的需求和特征不同，所以提供的服务接口的种类和数量也不同。在所有的支持模型中，可以划分出两类共性的类型：调度服务和执行服务。Aneka 定义了一个支持编程模型运行的参考模型，并抽象出调度和执行两个服务接口集合。此外，Aneka 还定义了基础接口来实现对新的编程模型的扩展支持。

##### 1. 调度

调度服务负责规划分布式应用的执行并将应用所包含的任务分配给节点。调度服务同时构成了几个其他基础服务和构造服务子集的调用结合点，例如资源分配、预留、记录、汇报。调度服务组件的共同任务如下：

- 节点映射任务。
- 失败任务的重新调度。
- 任务状态监控。
- 应用状态监控。

153

Aneka 并不提供集中式的调度引擎，每个编程模型拥有自己的调度服务接口，以便和中间件中的其他服务进行协作。如上文所述，协作服务主要属于图 5-2 所示框架中构造和集成服务接口层。为不同的编程模型设计不同的调度引擎给调度和资源分配机制的设计带来了很大的灵活性，但同时也需要对共享资源的使用进行精确设计。在调度过程中，对以下情况需要进行适当管理：多任务同时发送给同一节点，没有预留申请的任务发送给预留节点，接受任务的节点没有安装预先所需的服务内容。Aneka 基础服务提供了有效的信息收集接口来避免上述情况，但是应用运行平台并不具有检测和纠正上述情况的能力。Aneka 目前的设计机制是使每个调度引擎完全相互隔离，旨在需要时可以使用其他服务接口，以此来保证任意时刻在每个节点上对每个编程模型只有一个任务在执行，但应用的运行并不是互斥的，除非使用资源预留机制。

## 2. 执行

执行服务控制应用中单个任务的执行，负责设定执行任务的运行环境。调度服务实例运行时，每个编程模型都有自身需求，但在所有的编程模型中可以提取出一些共性操作：

- 从调度器获取任务后的分发。
- 取回所需的输入文件。
- 任务的沙盒运行。
- 任务执行完毕后输出文件的提交。
- 执行故障管理（例如抓取有效的上下文信息进行故障鉴别）。
- 性能监控。
- 任务打包并回传给调度器。

相对于调度服务，执行服务构成了一个自成体系的单元。执行服务需要处理的信息较少并且只需要和存储服务、本地部署和监控服务接口交互。Aneka 为执行服务提供了集成上述所有服务的关联接口集合，目前已有两个编程模型实现了专门的关联接口集合。

应用服务构成了对云中编程模型的运行支持，目前所支持的编程模型如下：

- 任务模型。包含很多任务划分应用和计算应用。应用被划分为相互独立的任务的集合，任务的执行可以按任意排序进行。
- 线程模型。对分布式框架下的经典多线程编程模型进行扩展支持，使用线程抽象模型来装载远端执行的程序模块。
- MapReduce 模型。MapReduce 在 Aneka 上的实现。
- 参数化模型。任务模型应用的一类实例，将多组参数集合输入任务实例，每组参数代表了选定域中一个具体的点。

154

其他编程模型正处于开发和测试阶段，其中包括数据流模型 [56]，信息传递接口和 Actor 模型。

## 5.3 构建 Aneka 云平台

Aneka 是一个开发分布式云应用的平台。作为一个软件平台，它需要部署在相应的硬件设施上，同时硬件设施需要相应的管理组件。构建云平台是云管理者的主要任务之一，Aneka 支持公共云、私有云和混合云的多种部署方式。

### 5.3.1 基础设施组织

图 5-3 从基础设施的角度描述了 Aneka 云平台，为不同的部署模式提供了一个参考模型。管理控制台作为核心角色执行所有需要的管理操作。云部署的一个基本元件是仓库集合，仓库存储了安装和部署云平台基本服务所需的所有库，这些库构成了用于节点管理和容器程序的软件镜像。仓库可以通过传输协议通道（HTTP、FTP、通用文件共享等）来共享库。管理平台可以管理多个仓库并为具体的部署选择一个合适的仓库。设备部署的过程是为一系列节点安装节点管理器，也称为守护进程。守护进程是远程管理服务实例的一部分，以实现容器实例的部署和管理。容器的集合构成了 Aneka 云系统。

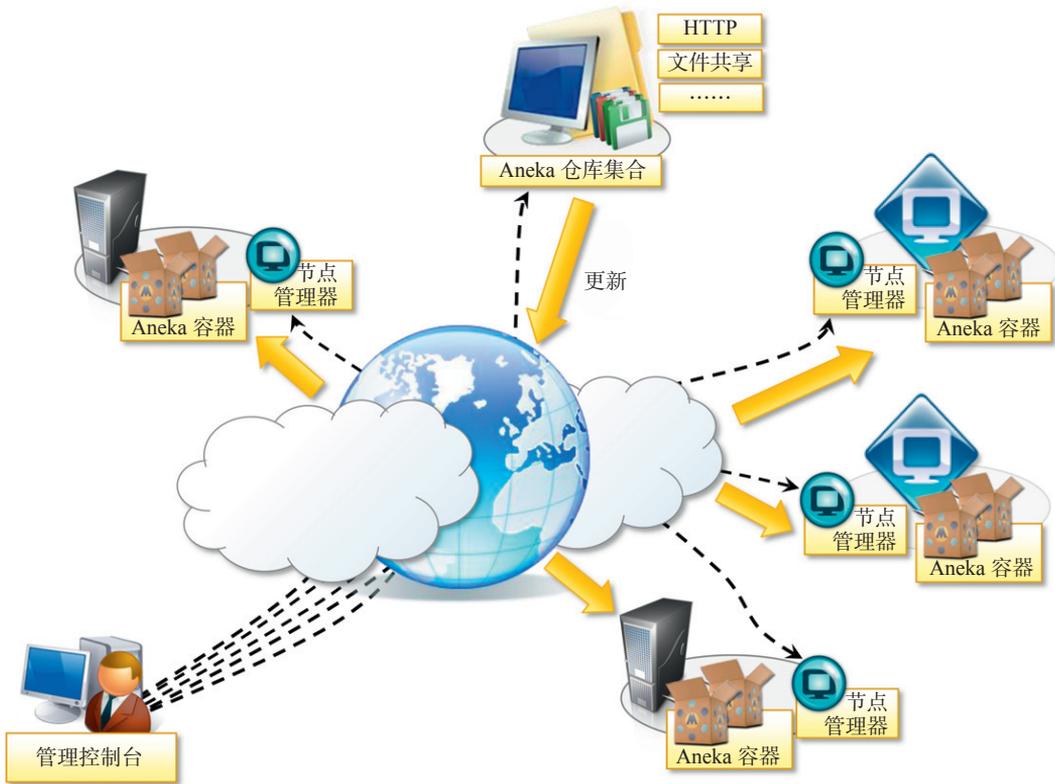


图 5-3 Aneka 云平台基础设施概览

从基础设施的角度而言，只要可以通过网络连接和远程管理接入节点，那么对物理节点和虚拟节点的管理就是一致的。而在虚拟化实例的动态分配过程中，容器由包含 Aneka 服务组件的打包资源镜像所创建，动态分配只需要将资源镜像配置给实例，以将实例加入到云环境中。容器和守护进程的安装也可以采取简化模式，这主要取决于虚拟资源的生命周期。

### 5.3.2 逻辑组织

云的逻辑组织非常多样化，主要取决于容器实例上的配置选择。常见的模式是采用主从配置，并另外划分存储节点，如图 5-4 所示。

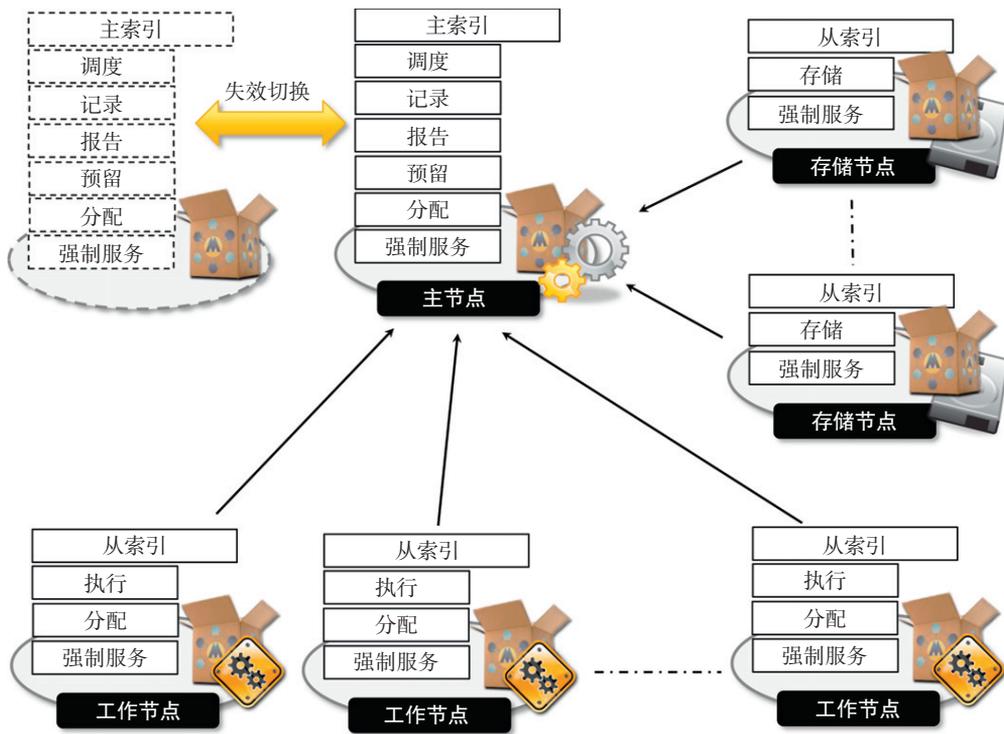


图 5-4 Aneka 云逻辑组织

主节点的服务通常具有一份拷贝，并提供对云内的智能管理。将一个节点设置为主节点需要在索引服务（成员目录）中将该节点配置为主节点模式。多拷贝的服务除了需要强制部署的以外，一般应配置到从节点。主节点的常规服务配置如下：

- 索引服务（主拷贝）。
- 心跳服务。
- 日志服务。
- 预留服务。
- 资源配置服务。
- 记录服务。
- 报告和监控服务。
- 编程模型对应的调度服务。

156

主节点通过一个关系型数据库（RDBMS）访问设备的运行状态数据。实际上，所有的调度服务都在主节点内完成。调度服务接口共享关系型数据库中的应用模块，以提供一种容错机制。主节点的配置可以被保存在几个其他节点上，通过故障转移机制提高系统的可靠性。

工作节点构成了 Aneka 云的处理单元，主要是为支持应用执行而配置相关服务，包括强制部署服务和具体编程模型的执行服务。常规配置如下：

- 索引服务。
- 心跳服务。
- 日志服务。

- 部署服务。
- 监控服务。
- 编程模型对应的执行服务。

157

通过选择不同的执行服务来使用不同的工作节点，可以实现负载均衡以及为一些具体应用保留相关节点。

存储节点为应用提供了优化存储支持，在强制服务和常规服务以外提供了存储服务接口。存储节点的数量取决于对工作负载的预测和应用对存储资源的消耗。存储节点主要配置在有足够硬盘空间可以容纳大量文件的设备上。存储节点的常规配置如下：

- 索引服务。
- 心跳服务。
- 日志服务。
- 监控服务。
- 存储服务。

实际使用中，如果没有数据传输需求，存储节点可能只有一个。在小型部署案例中，没有必要设置一个独立的存储节点，因此存储服务可能被安装和运行在主节点上。

所有节点都在主节点中进行注册，并且可以透明地关联到所有故障转移备用设备上，以保证设备的高可靠性。

### 5.3.3 私有云部署模式

私有云部署模式主要由本地物理机资源和提供资源池节点（多数情况下经过了虚拟化）访问的设备管理软件构成。这种情况下，Aneka 云需要在异构设备（例如桌面设备、集群和工作站）的资源池上构建。这些异构设备可以被划分为不同的组，云平台可以根据应用的需求来使用这些资源。此外，通过使用资源配置接口，可以从一些开源分布式虚拟化系统（如 XenServer、Eucalyptus 和 OpenStack）中获取虚拟节点资源。

图 5-5 显示了私有 Aneka 云的常规部署。在系统负载可以预测且本地虚拟机管理器可以处理超出预测负载的资源请求时，这种部署是可行的。大多数 Aneka 云中的节点由物理机

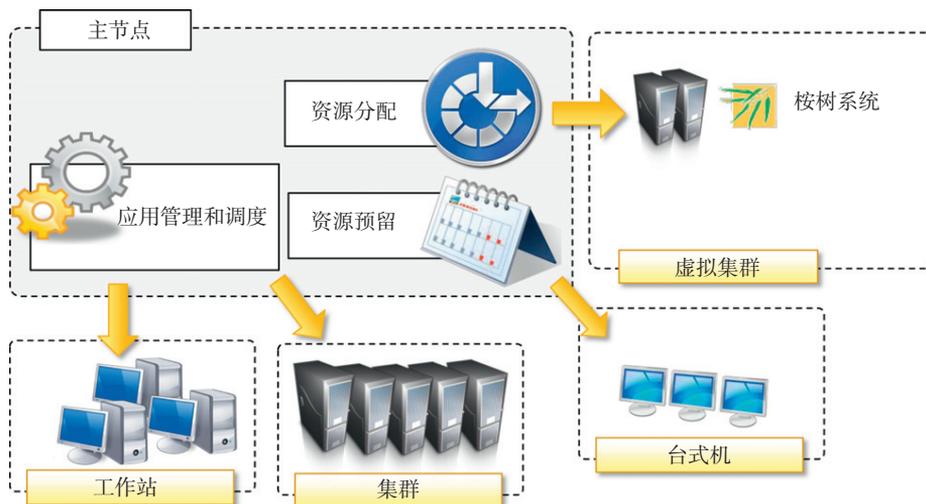


图 5-5 私有云部署

构成，具有很长的生命周期和静态配置并且不需要经常进行重新配置。针对私有云环境中物理机的不同属性，可以通过资源预留服务接口制定相应的资源管理和使用策略。例如，需要在白天进行办公自动化工作的台式机可以在标准工时以外执行分布式任务。工作站集群可能具有一些支持应用执行的既有软件，因此更适合有专属需求的应用。

### 5.3.4 公共云部署模式

158 在公共云部署模式下，Aneka 的主节点和工作节点全部部署在由一个或多个 IaaS 服务商（如亚马逊 EC2、GoGrid）提供的虚拟化设备上。所有节点都被事先提供以便静态部署。这种部署方式与在没有动态资源扩展能力的物理设备上安装 Aneka 是一致的。有趣的是，由于 IaaS 服务商的弹性服务特征，云的创建是完全动态化的。图 5-6 显示了这一场景。

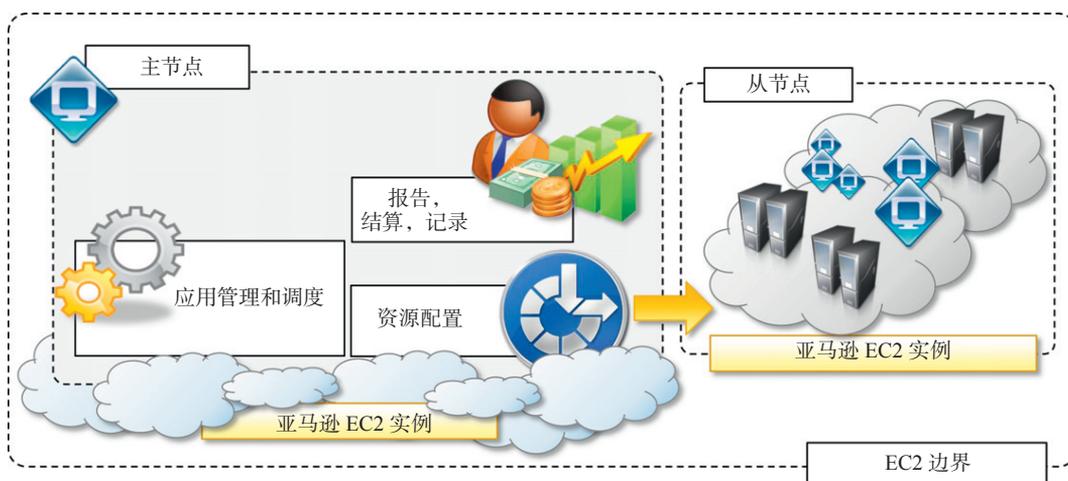


图 5-6 公共云部署

图 5-6 所示的部署是在单一 IaaS 服务商所提供的设备环境下进行的，目的在于避免不同服务商之间的数据迁移。服务商之间的数据迁移成本很高，同时部署在一个服务商的环境中网络性能更好。公共云环境下 Aneka 可以部署在单一节点上，完全使用动态分配接口按需求弹性调整设备规模。资源分配服务扮演了重要角色，可以将不同的镜像和模板配置给实例。主节点中必须包含的服务还有记录和报告服务。它们向用户和应用提供资源的使用细节，在多用户云环境中起着十分基础的作用，而用户依据它们对资源的消耗情况进行结算。

159 动态实例的分配将主要应用于工作节点的分配，在亚马逊 EC2 服务中，不同的镜像包含了不同的硬件资源，工作容器可以从中选取并进行配置。具有计算或存储需求的应用将会向调度器传递附加信息来触发合适的分配请求。应用执行并不是唯一使用动态实例的情况，任何使用弹性扩展的服务都会使用到动态分配机制。存储服务是另一个实例，在多用户云中，多种应用可能需要存储服务的支持，有可能造成瓶颈或达到存储节点的配额限制。动态分配可以通过增加云的计算能力来解决这一问题。

由于在不同服务商间进行数据迁移的巨大成本，目前在不同服务商上部署云还不太可能，但是这可能成为联盟 Aneka 云的蓝图 [58]。在这种场景下，通过相应协议和更优惠的价格，资源可以在服务商之间实现共享和租用。资源分配服务策略可以区分不同的资源供应商，通过对不同的 IaaS 服务商的映射提供需求分配的最佳解决方式。

### 5.3.5 混合云部署模式

混合部署模式是 Aneka 最常见的部署模式，在很多实例中都需要一个易用的计算框架来处理应用的计算需求。该框架可以是一个 Aneka 部署的可按需求动态扩展的框架。这种部署如图 5-7 所示。

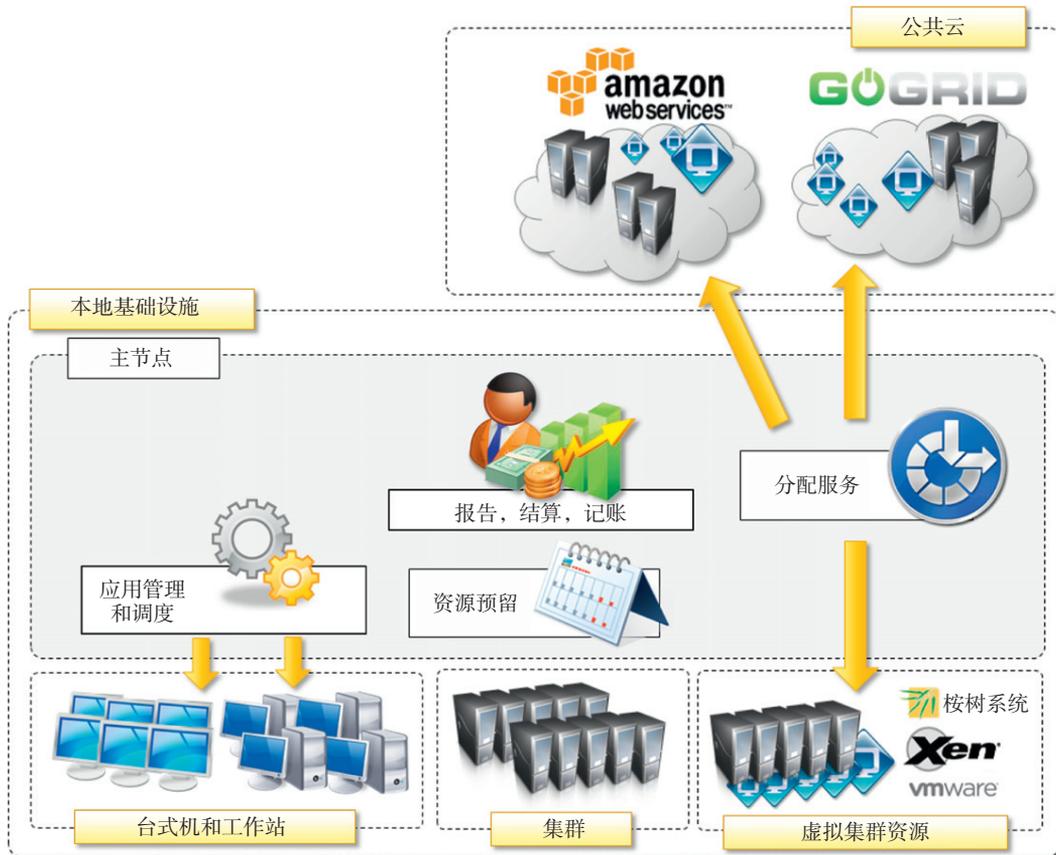


图 5-7 混合云部署

这种部署是最完整的模式，能够使用 Aneka 框架的所有功能，包括：

- 动态资源分配。
- 资源预留。
- 工作负载划分。
- 记录、监控和报告。

此外，如果在本地能够管理一些虚拟机资源，则可能实现对资源的充分利用，从而降低应用执行的费用。

在混合部署场景下，可以为不同用途提供不同的异构资源。正如私有云部署中所介绍的，桌面设备可以在办公时间以外预留给低优先级的工作负载，大多数应用在工作站和集群中运行，这些节点始终和 Aneka 云相连接。任何增加的计算能力需求都可以首先被本地虚拟化设备所处理，如果需要更多的计算能力，可以使用外部 IaaS 服务商的资源。

混合云部署与公共云部署的区别是利用了多元化的资源供应商来提供虚拟资源。由于部

[161]

分设施是本地的，数据迁移到外部 IaaS 服务商的设备上产生的花费是不可避免的。因此在处理应用请求时要选择合适的方案。Aneka 中的资源分配服务提供了同时使用若干资源池以及设置最佳资源池配置策略的能力。上述特性可以简化用户策略的开发，更好地满足混合部署的需求。

## 5.4 云编程和云管理

Aneka 的初衷是提供一个执行分布式应用的可扩展中间件产品。对开发者和管理者而言，应用开发和管理是 Aneka 最主要的产品特性。为了简化上述过程，Aneka 向开发者提供了全面的可扩展的 API 集合，向管理者提供了有力的和直观的管理工具。开发者的 API 主要集中在 Aneka SDK 中，管理工具主要通过管理控制台提供。

### 5.4.1 Aneka SDK

Aneka 提供 API，用于基于现有编程模型开发应用、实现新的编程模型以及开发新的云平台集成服务接口。应用开发主要关注应用现有特性和使用中间件服务接口，而新的编程模型和服务接口开发将会丰富 Aneka 的特性。SDK 通过应用模型和服务模型的方式来提供对编程模型和服务接口的编程支持。前者与应用开发和新编程模型开发有关，后者则通过服务接口开发定义通用框架。

#### 1. 应用模型

Aneka 通过提供抽象编程模型实现对云中分布式执行的支持。编程模型定义了开发者所使用的抽象以及程序在 Aneka 上运行所需的运行支持。应用模型表示适用于所有编程模型（代表 Aneka 平台上的分布式任务）的最小化 API 集合。根据需求和每个编程模型的特性，可进一步定制应用模型。

图 5-8 显示了编程模型所定义的组件，Aneka 平台运行的每个分布式应用都是 `ApplicationBase<M>` 类的实例，M 定义了应用管理器的类型。应用类构成了开发者视角下的分布式应用，而应用管理器是类的内部成员，用来和 Aneka 云进行交互以监视和控制应用的执行。应用管理器是编程模型实例中的第一个元素，并根据编程模型的不同选择来不对不同的管理器进行配置。

[162]

无论使用哪种编程模型，分布式应用都可以在接收端被分解为一系列任务。Aneka 将应用进一步划分为两类：基于用户任务的应用和基于运行平台任务的应用。这两类应用对应了不同的应用基类以及不同的应用管理实现。

第一类应用最为常见并关联了几种编程模型：任务模型、线程模型以及参数化模型。该类应用由一系列用户提交的工作单元组成，每个单元都是 `WorkUnit` 类的子类。每个工作单元具有输入和输出文件，文件的传输由运行环境进行透明管理。工作单元的具体子类取决于编程模型（任务模型对应 `AnekaTask` class，线程模型对应 `AnekaThread` Class）。用户任务的应用都是 `AnekaApplication (W, M)` 的子类或实例。W 表示 `WorkUnit` 类的子类（多态性），M 表示 `IManalApplicationManager` 接口的实现类。

第二类应用包含了 `MapReduce` 以及其他基于运行平台的应用，这种情况下工作单元没有通用的类型，应用开发者所使用的具体类型取决于编程模型的需要。例如在 `MapReduce` 模型中，开发者用两种函数表示它们的分布式应用，`map` 和 `reduce`。因此，`MapReduceApplication` 类为 `Mapper<K,V>` 和 `Reducer<K,V>`，提供相应接口以及应用所需的输入文件。其他编程

模型可能具有不同的需求和接口。因此，除了 ApplicationBase<M> (M 表示 IautoApplicationManager)，这一类别中没有其他通用基类。

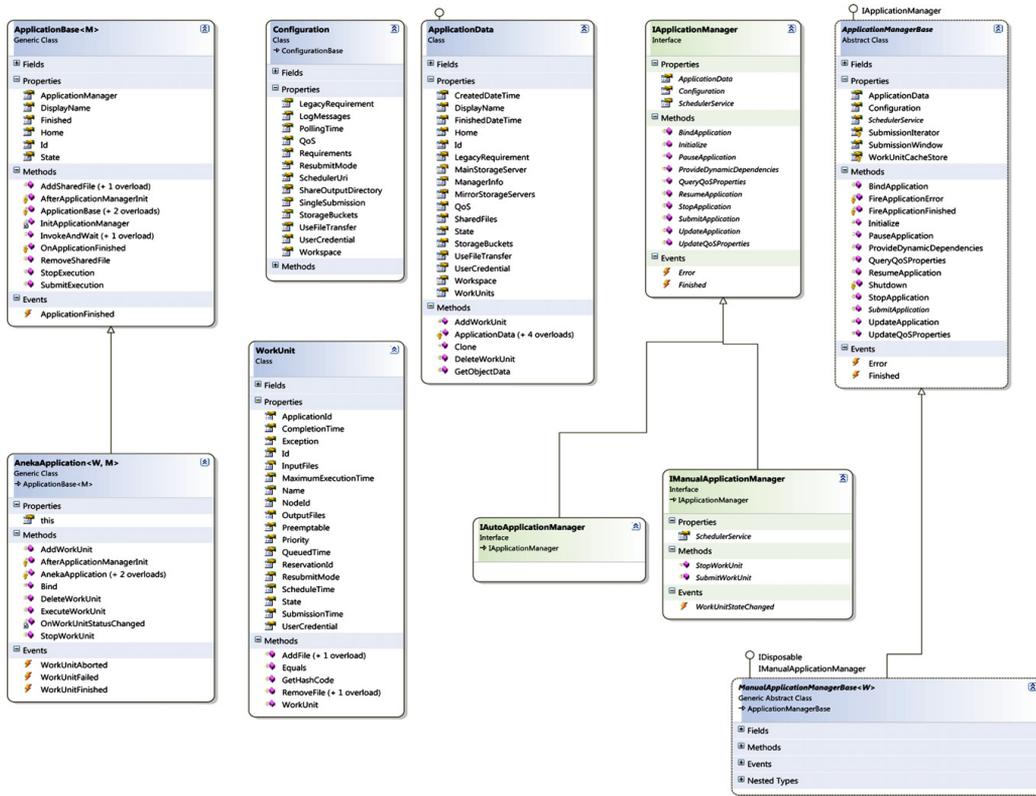


图 5-8 Aneka 应用模型

一系列附加类组成了对象模型，在这些类中，最值得关注的是 Configuration 类，它用来初始化应用并配置其行为。以及 ApplicationData 类，它包含应用的执行信息。

表格 5-1 概括了目前 Aneka 可支持的应用模型的特征以及对应的编程模型。编程模型是可以扩展的，因此可以利用表格中的基类构造新的编程模型。编程模型扩展可以通过增加（具体化）现有编程模型的实现特征，或者使用基类来定义新的编程模型和抽象来实现。例如，参数化模型是任务模型的扩展，在 Aneka 云应用管理环境中已经实现。此类模型向终端用户提供特定接口，用户只需定义相关模板任务和参数。

163  
?  
164

表 5-1 Aneka 应用模型特征

类别	描述	基本应用类型	工作单元	编程模型
手动	工作单元由用户产生，并通过应用提交	AnekaApplicaton <W, M> IManualApplicationManager <W> Manual ApplicationManager <W>	是	任务模型 线程模型
自动	工作单元由运行中的基础设施产生，并在其内部进行管理	ApplicationBase <M> IAutoApplicationManager	否	参数化模型 MapReduce 模型

## 2. 服务模型

Aneka 云服务模型定义了实现一个云服务所需的基本要求。容器定义了加载服务的运

行环境。每项加载在容器中的服务必须加载一个 `IService` 接口，该接口呈现了以下方法和属性：

- 名称和状态。
- 控制操作，例如开始、停止、暂停和继续。
- 消息处理方法。

与终端用户直接交互的服务可以直接提供给用户，例如资源分配和预留服务。除了使用容器来开启和关闭服务的控制操作外，服务的核心逻辑都包含在消息处理机制中，位于 `HandleMessage` 方法下。每项需要使用服务的操作由具体消息所触发，操作结果也通过消息回执给调用者。

图 5-9 描述了容器中每个服务实例的生命周期。阴影圆圈表示临时状态，空白圆圈表示稳定状态。服务实例初始化后可以进入未知或初始化状态，一种在容器配置阶段触发容器中相应构造器的状态。一旦容器启动，容器会对每项服务调用启动函数，之后服务实例会进入启动状态。启动过程完成后进入运行状态。在容器运行周期内，服务将一直处于运行状态。运行状态也是服务唯一能够处理消息的状态。如果在服务启动过程中发生了异常，服务实例将会返回未知状态，并返回一个错误标识符。

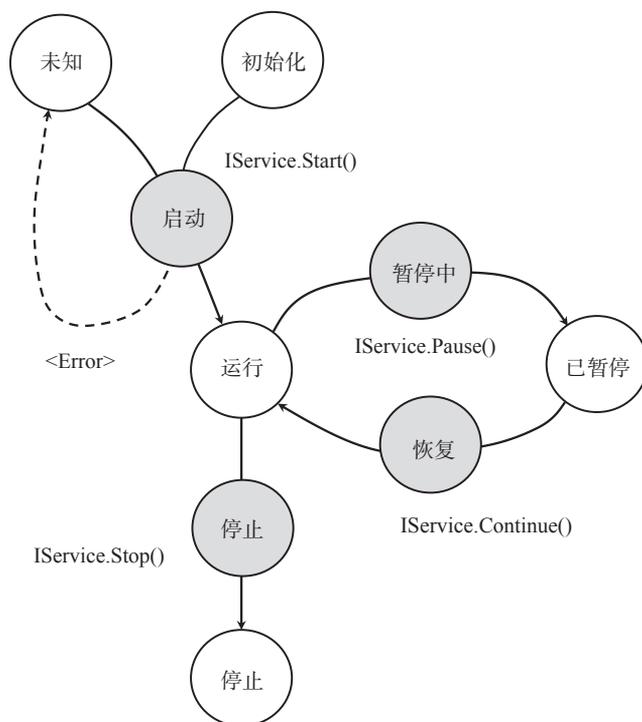


图 5-9 服务生命周期

服务在运行过程中可以使用暂停和继续方法使服务暂停和继续。如图 5-9 所示，服务实例首先进入暂停中状态，然后进入已暂停状态，再之后进入恢复状态并恢复其活动直至返回运行状态。不是所有服务都需要支持暂停 / 继续操作。目前框架中并不是所有服务都提供上述功能。

当容器注销时，容器对每项服务调用停止方法，并进入到最终的停止状态。所有预先分

配的资源将会被释放。

Aneka 提供了一个默认基类来简化服务的实现，同时对服务开发者设定了一些引导规则以便所开发的服务能够与 Aneka 兼容。特别是定义了可扩展的 Servicebase 类来实现服务，该类是框架中一些服务的基类并提供了一些内置特性：

- IService 接口所呈现的基本属性的定义。
- 对访问能力和状态控制相关操作的实现。
- 传递服务用户信息的内置框架。
- 支持服务监控。

开发者利用系统提供的定义控制操作行为的模板方法，实现自定义的消息处理机制，并提供对应的服务客户端。

Aneka 使用了一个强类型的信息传递通信模型，每项服务定义自己的消息体系，不同消息的设定是独占的。因此，新服务接口的开发者也需要定义所开发服务使用的消息类型，以便和其他服务和客户端通信。每个消息类型都是 Message 基类的集成子类。Message 类定义了如下特性：

- 源节点和目标节点。
- 源服务和目标服务。
- 安全凭据。

每个信息类型还包含附加信息。在 Aneka 平台中，信息传递应用广泛。对于应用直接使用的服务，它们会提供一个服务客户端，提供面向对象的操作接口。Aneka 提供了一个易用平台，可以通过中间件将服务客户端动态注入应用。继承自 ServiceBase 类的服务已经支持了上述特征，只需要为服务客户端类定义接口和具体实现。服务客户端在将 Aneka 服务集成进已有应用方面十分方便，已有应用不需要分布式应用的支持或者接入额外的服务接口。

Aneka 在服务配置方面也提供了丰富的支持。开发者可以定义 editors 类和 configuration 类，通过容器所需的工作流将服务配置信息集成到管理工具中。

## 5.4.2 管理工具

Aneka 是一个纯 PaaS 实现，使用虚拟或物理设备资源进行部署。因此，基础设备管理，以及在设备上安装逻辑云的相关设备管理，是管理接口层的重要特征。该层也包括了管理云中服务和应用实例的能力。

### 1. 设备管理

Aneka 使用虚拟或物理设备资源来部署云平台。虚拟硬件总体上通过资源分配服务进行管理，应用根据需求获取资源。物理硬件通过 PAL 层的管理 API，由管理控制台直接管理。管理的内容主要是物理硬件的分配和 Aneka 在硬件上的远程安装。

### 2. 平台管理

设备管理提供了云部署环境上的一个基础接口层。云创建的过程就是将一系列服务部署在物理设备上，以实现容器的安装和管理。一系列互联的容器定义了应用的执行平台。平台管理的主要内容是云的逻辑组织和框架。Aneka 可以将可用的硬件资源划分到若干云中作为不同用途使用。服务接口实现了 Aneka 云的核心特征，管理接口层提供了一些服务的操作接口，例如云监控、资源分配和预留、用户管理、应用概要分析。

165

166

### 3. 应用管理

应用定义了用户对云的贡献。管理 API 提供监控和分析功能，帮助管理者记录用户和应用对资源的使用。在以资源使用作为用户结算依据的云环境下，这是十分重要的功能。  
[167] Aneka 提供了对应用执行和资源使用生成宏观和微观分析信息的能力。

上述特征通过主管理控制台的管理工作室来实现。

## 本章小结

本章介绍了 Aneka，一个支持云应用编程的平台。Aneka 是一个纯 PaaS 实现，构成了一个能够在异构资源（台式机、集群、公共虚拟机）上创建云平台的中间件产品。

Aneka 框架的一个核心特征是可配置运行环境，可以创建基于服务的应用执行中间件。平台的一个基础元素是容器，代表了云的部署单元。容器装载了一系列服务，定义了中间件的处理能力。Aneka 中间件的基础服务如下：

- 构造服务，支持监控、资源分配、硬件分析和成员管理。
- 基础服务，支持存储、资源预留、结算、记录和报告。
- 应用服务，支持调度和执行。

从编程应用的角度，Aneka 提供对不同编程模型的支持，允许开发者通过不同抽象模型扩展分布式应用。该框架目前支持三种模型：相互独立任务的模型、多线程模型和 MapReduce。

Aneka 平台是可扩展的，通过提供应用模型和服务模型来实现扩展，将新的服务接口和编程模型集成到系统内部。

## 习题

1. 请简单描述 Aneka 的主要特点。
2. Aneka 容器是什么？它的用处是什么？
3. Aneka 容器承载的服务类型有哪些？
4. 描述 Aneka 的资源分配功能。
5. 描述 Aneka 的存储架构的实现。
6. 什么是编程模型？
7. 列举 Aneka 支持的编程模型。
8. Aneka 的基础结构组成组件有哪些？
9. 讨论 Aneka 云的逻辑组织结构。
10. 工作节点承载了哪些服务？
11. 讨论 Aneka 云的私有部署。
- 168 12. 讨论 Aneka 云的公共部署。
13. 讨论混合部署中动态资源分配的角色。
14. Aneka 为开发提供哪些设施？
15. 讨论 Aneka 应用模型的主要特征。
- 169 16. 讨论 Aneka 服务模型的主要特征。
- 170 17. 讨论 Aneka 管理工具在基础设施、平台和应用中的特征。