**阿里云体系架构**

**2017221105020**

**牟钰潇**

**云计算-指导老师：张帆**

1. **飞天体系架构**

**1.1飞天相关简介**

飞天操作系统，（项目代号“Apsara”）是阿里云公司自主研发的分布式计算平台。

它的主要设计目标是通过构建一套综合性的软硬件系统，使得用户或者是应用系统可以像使用一台计算机一样便捷地使用数以千计地服务器地存储资源和计算资源。

它通过对软件系统地合理分层，使得“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”可以共享一个物理集群的计算，存储和网络资源，以极大幅度地提升大规模集群的使用效率。

飞天体系架构主要有：分布式系统底层服务（主要提供分布式环境下所需要的协调服务（代号女娲）、远程过程调用（夸父）、以及提供系统安全的钟馗模块），分布式文件系统（主要提供一个海量的，可靠的，可扩展的数据存储服务，将集群中的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），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和强调处理数据吞吐量的离线任务），集群监控和部署（对集群的状态和事件进行监控，对异常事件产生警报和记录；为运维人员提供整个飞天系统以及上层应用的部署和配置管理，支持在线集群的扩容和应用服务的在线升级）。



飞天操作系统，（项目代号“Apsara”）是阿里云公司自主研发的分布式

计算平台。

它的主要设计目标是通过构建一套综合性的软硬件系统，使得用户或者

是应用系统可以像使用一台计算机一样便捷地使用数以千计地服务器地

存储资源和计算资源。

它通过对软件系统地合理分层，使得“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”可以共享一个物理集群的计算，存储

和网络资源，以极大幅度地提升大规模集群的使用效率。

飞天体系架构主要有：分布式系统底层服务（主要提供分布式环境下所

需要的协调服务（代号女娲）、远程过程调用（夸父）、以及提供系统安

的钟馗模块），分布式文件系统（主要提供一个海量的，可靠的，可扩

的数据存储服务，将集群中的各个节点的存储能力聚集起来，并能够

屏蔽软硬件故障，为用户提供不间断的数据访问服务），任务调度（为

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和强

**1.2分布式基础架构**

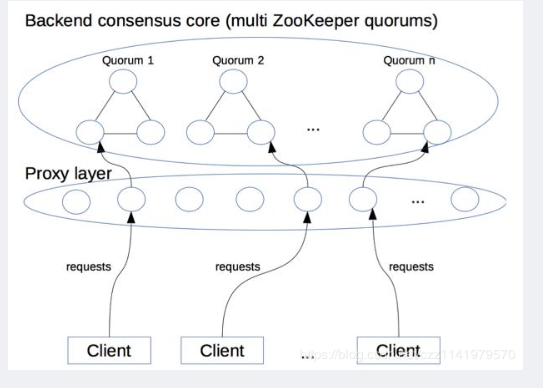
**1.2.1命名服务——女娲**

女娲系统为飞天平台提供了高可用的协调服务，是整个飞天系统的一个核心服务，它的作用采用类似文件系统的树命名空间来让分布式的进程相互协调工作。

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。

女娲基于类Paxos协议，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，根目录的名字是“/”，所有目录的名字都是以“/”结尾的。



女娲系统为飞天平台提供了高可用的协调服务，是整个

飞天系统的一个核心服务，它的作用采用类似文件系统

的树命名空间来让分布式的进程相互协调工作。

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。

女娲基于类Paxos协议，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。

女娲系统与Google的Chubby和Hadoop的ZooKeeper系统的功能与实现相似。

女娲具有以下特点：

·女娲服务存储了大量的键对

·女娲支持Publish/Subscribe模式

·女娲还可以用来实现复杂均衡

与有类似功能的开源软件相比,女娲在性能,可扩展性,和可运维性上有明显优势。目前,女娲在阿里云支撑着大数据计算，弹性计算，表格存储，对象存储等众多重量级的云产品。

**1.2.2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（RPC）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。

而夸父是飞天平台中负责网络通信的组件，它提供了一个RPC的接口，简化编写基于网络的分布式应用。在异步调用时，RPC函数调用时不等接收到结果便会立即返回，用户必须通过显式调用接收函数取得请求结果。

与用流传输的TCP通信相比，夸父通信是以消息为单位的，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同时支持异步和同步的远程过程调用形式。

在同步调用时，RPC函数调用时便会等待，知道接收到结果后菜返回。在实现中，同步调用是通过封装异步调用来实现的。

夸父的设计目标是提供高可用（7 x 24小时）、大吞吐量、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

**1.2.3安全管理——钟馗**

钟馗是飞天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。

用户的身份认证是基于密钥机制的。

用户对资源的访问控制是基于权能机制进行授权的。

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。

**1.3分布式文件系统——盘古**

**1.3.1分布式文件系统——盘古简介**

盘古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。

盘古与Google文件系统（GFS）和Hadoop的HDFS的设计目标有一致的部分，都是讲大量廉价的机器的存储资源聚合在一起，为用户提供大规模，高可靠，高可用和可扩展的存储服务，是集群操作系统中一个重要的组成部分。除此之外，盘古还能够很好的支持在线应用的低延时需求，这是GFS和HDFS所不具备的。

**1.3.2特点**

·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。

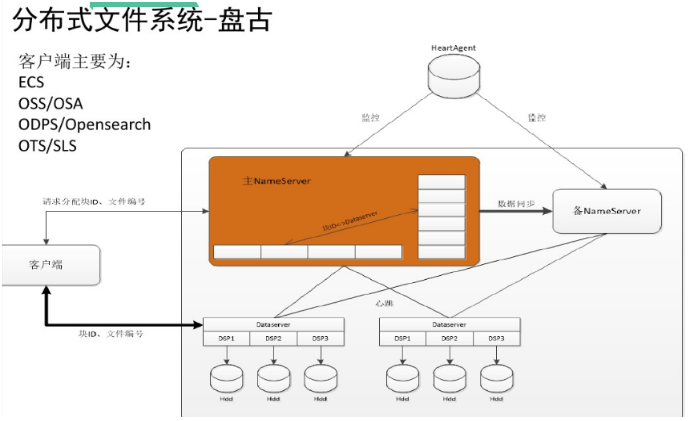
高可扩展性：保证系统的容量能够通过增加机器的方式得到自动扩展，下线机器存储的数据能够自动迁移到新加入的节点上。

**1.3.3相关设计**

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。



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。

**1.3.4相关优化**

盘古虽然优点非常显著，但是仍然避免不了需要优化，OSS、OTS等要求低延时的数据读写，ESC要求随机写的能力。

日志文件则可以通过多种方法优化延时：设置高优先级，Client直接写多份拷贝、写成功不经过Master确认等

随机访问文件：运行用户随机写等。

**1.4任务调度——伏羲**

**1.4.1任务调度——伏羲简介**

伏羲（Fuxi）是飞天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**。

**1.4.2设计与实现**

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，达到在保障公平的前提下，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、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。（如果熟悉spark技术的朋友，有没有感觉像是spark中的master和worker进程）

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集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ServiceMaster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JobMaster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ServiceMaster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ServiceWorker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。

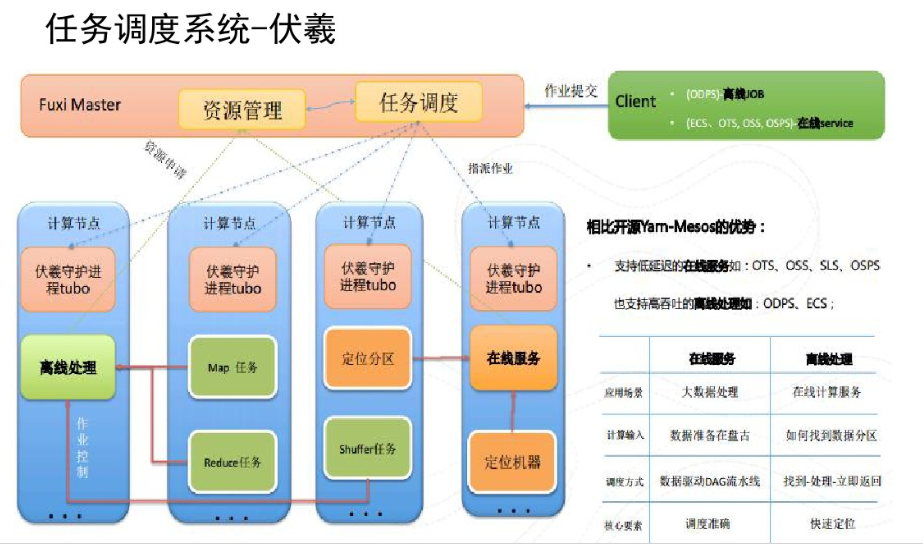
在服务运行的过程中，每个Service的数据分片的数目和内容都是可以动态变化的，应用程序可以根据实际需要对数据分片动态地进行加载(Load)、卸载(Unload)、分裂(Split)和迁移(Migrate)等操作。

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JobMaster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。

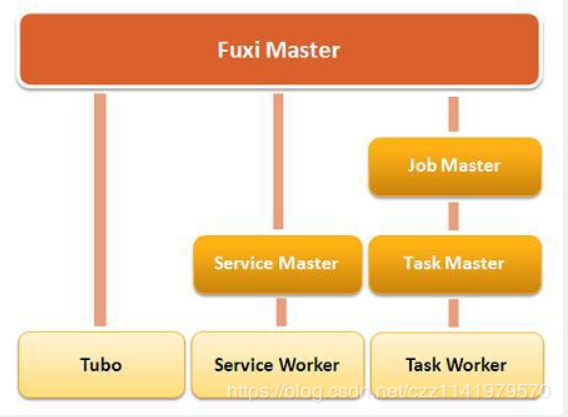


土伯(Tubo)是部署在每台由伏羲管理的机器上的后台进程，负责收集并向伏

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ServiceMaster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JobMaster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。



**1.5集群监控——神农**

**1.5.1集群监控——神农简介**

神农（Shennong）是飞天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。

**1.5.2相关设计**

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。

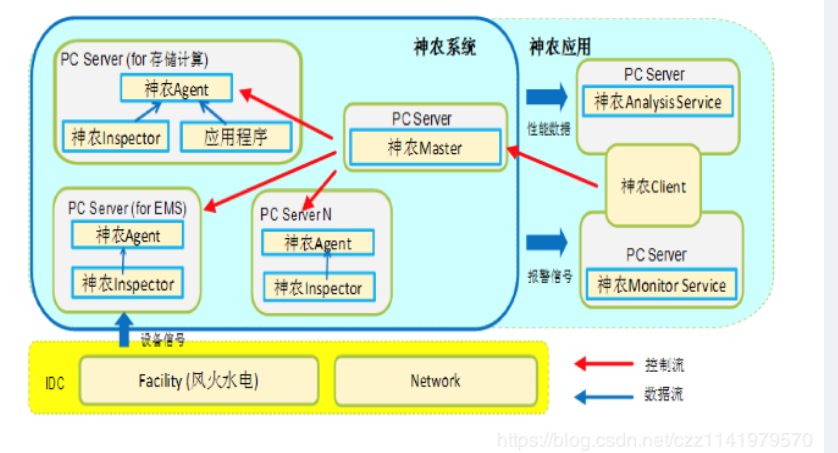
Agent：是部署在每台物理机器的后台程序。Agent负责接受来自应用和Inspector写入的信息。Agent启动后，会立刻向Master注册自己，并根据Master发来的订阅(Subscription)命令执行相应的信息采集、过滤、聚合和处理操作。目前神农Agent处理的数据分为两类：事件类数据(如应用程序故障和报警)和数值类数据(如当前应用的性能计数、机器I/O吞吐量等)。

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。



**1.6集群部署——大禹**

**1.6.1集群部署——大禹简介**

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，以及为其它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。

**1.6.2结构设计**

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。

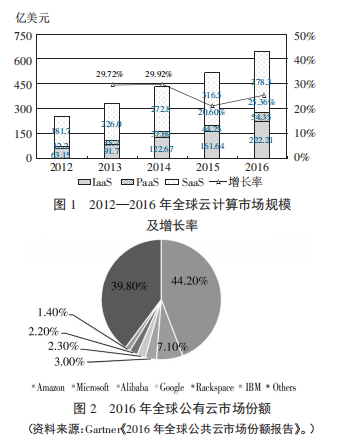
集群配置数据库负责存放和管理所有部署了飞天的集群的配置信息，包括集群中每个节点承担的角色、各个模块的软件版本、各个模块的基本参数配置等。同时，数据库中还记录了部署或升级时每个节点的任务执行状态，保证了在部署或升级时少量不在线节点可以在重新连线后进行自动修复。

 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。

**1.7发展趋势**

2016 年全球云计算市场平稳增长，市场总体规模为 522.4 亿美元，增速上比 2015 年有所上升，达到 25.36%。如图 1 所示，从细分市场来看：IaaS 市场贡献 222.21 亿美元，连续保持稳定增长；PaaS 市场贡献较少为 54.35 亿美元，增速有所放缓但数据库服务和商业智能平台服务增长较快；SaaS 市场贡献最多将近 400 亿，是全球公有云市场的最大构成部分。如图 2 所示，亚马逊是公有云市场的明显领导者，其 44.2%的市场份额决定了绝对的市场统治地位，远远高于随后的微软、阿里巴巴和谷歌三者份额总和。份额较大的几家服务商公司大部分来自美国，作为云计算的“先行者”，美国在市场中处于明显优势地位；欧洲作为云计算市场重要组成部分，近年来增长放缓但对全球市场有一定影响，如果这样的态势继续下去，将进一步拉大与美国之间的差距；以中国为代表的云计算新兴国家正在高速增长，2016 年阿里云“挤出”谷歌云成功入围全球公有云市场份额前三甲，被业内视作标志性的事件，表明中国云计算有巨大的潜力可挖掘，预计未来市场将进一步扩大。



1.“云+AI”成为新趋势。云计算是数据处理的天然土壤，其大规模并行和分布式计算能力可以帮助减少 AI（Artificial Intelligence，人工智能）的计算成本。目前来看，作为提升数据处理效率有力武器的 AI 已经是云计算厂商们势在必得的市场，谷歌、亚马逊、IBM 等行业巨头均已成立专门部门加入人工智能领域的“战争”，“云+AI”融合发展已成为新趋势。

2. 私有云市场蓬勃发展。传统 IT 厂商、技术创新企业聚集在一起，形成私有云众筹式发展的局面，它们借助开源社区和平台的力量，争取在云计算市场竞争中能与亚马逊、谷歌、微软等云服务行业巨头相抗衡，目前已成为云计算产业生态另一个核心的发展趋势。

3. 云安全出现潜在威胁。“云”的存在使得海量数据资源得以规模化和集中化，一旦数据中心、网络线路等物理设施遭到人为破坏或自然故障，将对云安全造成不可估量的影响。这对云服务商的运维水平来讲无疑是十分艰巨的挑战。