

云计算架构介绍

白皮书

第 1 版，2009 年 6 月

摘要

云计算可望提高应用程序部署速度、促进创新和降低成本，同时还增强经营敏捷性。Sun 抱持一种全面的云计算观点，因而可以支持各个层面，其中包括服务器、存储、网络 and 虚拟化技术，这些技术将云计算环境扩展到虚拟设备中运行的软件，而这些虚拟设备可用来在极少时间内成功汇编应用程序。本白皮书探讨云计算如何变革我们的设计、构建和提供应用程序的方式，以及企业在采纳并应用云计算技术时必须考虑的架构问题。

本页故意留空。

目录

引言.....	1
Sun 公司观点.....	1
云计算的性质	2
扩大已形成的趋势	2
将虚拟机作为标准部署对象	2
按需、自助、以使用情况付费的模式	2
通过网络提供服务	5
开放源软件的作用	5
云计算基础设施模式	6
公用云、专用云和混合云	6
云计算的架构层	9
云应用程序设计接口	11
云计算效益	11
缩短运行时间和响应时间	11
最大限度地减轻基础设施风险	12
降低入市成本	12
加快创新步伐	12
实现 IaaS 必须考虑的架构问题	13
不断发展的应用程序架构	13
变革架构的途径	13
变革应用程序设计	13
目标仍然相同	14
一致而稳定的抽象层	16
标准有助于解决复杂问题	16
松散耦合、无状态、原地失败 (Fail-in-Place) 计算	18
水平扩展	18
并行化.....	19
分割并征服	20
数据物理	21
数据与处理之间的关系	21
编程策略	21
合规与数据物理	22
安全性与数据物理	22
网络安全做法	23
Sun 公司与云计算.....	24
来自 Sun 社区的创新	24
社区与开放式标准	25
选择的重要性	25
选择云计算提供商	25
感谢.....	26

第 1 章 引言

什么是云计算 (Cloud Computing)? 每个人都有自己的看法。云计算可以是租用一台服务器或一千台服务器并在目前世界上最强大的系统上运行地球物理建模应用程序的能力。云计算也可以是租用一个虚拟服务器、在其上面加载软件、随意打开和关闭该虚拟服务器或克隆该服务器十次以满足突发工作负荷需求的能力。云计算可以存储并保护巨量数据, 而且这些数据只允许授权应用程序和用户进行访问。云计算可由建立一个包括 OS、Apache、MySQL™ 数据库、Perl、Python 和 PHP 的平台的云提供商提供支持, 该平台能够根据不断变化的工作负荷自动扩展。云计算还可以是在互联网上使用存储并保护数据同时提供服务 [包括电子邮件、销售能力自动化 (Sales Force Automation) 和报税 (Tax Preparation), 等等] 的应用程序的能力。云计算可以使用存储云 (Storage Cloud) 来保存应用程序、企业和个人数据。而且, 云计算还可以使用少数 Web 服务来集成照片、地图和 GPS 信息, 以便与客户 Web 浏览器中创建聚合 (Mashup)。

Sun 公司观点

Sun 公司采取一种全面的观点, 这种观点认为, 云的类型有很多种, 而且有很多不同的应用程序可以使用云来构建。由于云计算有助于提高应用程序部署速度, 有助于加快创新步伐, 因而云计算可能还会出现我们现在无法想像到的形式。然而, 永恒不变的是, Sun 公司是一家具有丰富经验的服务器、网络 and 软件技术提供商, 我们提供的这些技术均支持云计算。作为创造“网络就是计算机” (The Network is the Computer™) 这一短语的公司, 我们深信云计算就是下一代的网络计算。

云计算与以前的模式有什么区别? 就云计算这个短语而言, 它就是通过网络把信息技术当作服务来使用。我们将其定义为封装的、具有 API 且通过网络提供的服务。此定义同时包含把计算和存储资源当作服务使用。云计算首先以效率原理为基础, 此处所说的效率是指产生用来处理 80% 的使用案例的高级工具, 这样才能以一种惊人的速度创建和部署应用程序。

云计算可以由企业数据中心自己的服务器进行提供, 也可以由承担拥有基础设施的全部风险的云提供商提供。错误的观念认为资源是取之不尽的。尽管该领域还处于萌芽阶段, 但其模式已在信息技术 (IT) 行业引起极大反响。现在, 云计算的主要模式称为“把基础设施当作服务” (IaaS), 而且由于其突出特性, IaaS 模式成为本白皮书第一版本的焦点。

本白皮书探讨云计算的性质及其如何在变革世界各地的企业构建和部署应用程序的方式的同时扩大已形成的趋势。然后, 讨论云架构设计师 (Cloud Architect) 设计基于云的应用程序时必须考虑的架构问题。最后, 介绍 Sun 公司提供的支持云计算的技术。

第 2 章

云计算的性质

扩大已形成的趋势

云计算推动降低服务提供成本的已有趋势，同时提高部署服务的速度和敏捷性。它缩短了从设计应用程序架构到实际部署应用程序的时间。云计算把虚拟化、按需部署、网上服务提供和开源软件融合在一起。从一种观点看，云计算并非新生事物，因为它使用既有的方法、概念和最佳做法。而从另一种观点看，一切都是新的，因为云计算变革我们发明、开发、部署、扩展、更新、维护和支付应用程序以及运行应用程序的基础设施的方式。在本章中，我们考查上述趋势，以及这些趋势是如何成为云计算的性质的核心的。

将虚拟机作为标准部署对象

在过去几年时间里，虚拟机已成为一种标准部署对象。虚拟化进一步增强了灵活性，因为它把硬件概括到这样一个高度：在硬件上面，可以在不需要连接具体物理服务器的情况下部署和重新部署软件栈。虚拟化实现了一个动态数据中心，其中的服务器提供一个包含可根据需要使用资源的资源池，而且，其中的应用程序与计算、存储和网络资源的关系可动态变化，以适应工作负荷和业务需求。由于应用程序部署与服务器部署相分离，因而可以快速部署和扩展应用程序，而不必首先购置物理服务器。

虚拟机已成为流行抽象概念 — 和部署单位 — 因为它们是服务提供商和开发人员之间的最小公分母连接体。把虚拟机用作部署对象足以适应 80 % 的使用情况，而且这将有助于满足快速部署和扩展应用程序的需要。

虚拟设备 (包含软件的虚拟机，这些软件部分或全部地配置为执行像 Web 服务器或数据库服务器这样的特定任务) 进一步增强了快速创建和部署应用程序的能力。把虚拟机和设备作为标准部署对象组合在一起是云计算的关键特性之一。

计算云通常由存储云进行补充，存储云通过 API 提供虚拟化存储，而这些 API 为存储虚拟机映像 (Image)、用于诸如 Web 服务器的组件的源文件、应用程序状态数据以及一般业务数据，提供便利。

按需、自助、以使用情况付费的模式

云计算的按需、自助和以使用情况付费的性质也是已有趋势的一种延伸。从企业的观点看，云计算的按需性质有助于支持服务水平目标的性能和容量方面。云计算的自助性质使机构可以创造根据工作负荷和目标性能参数进行扩展和收缩的弹性环境。而且云计算的按使用情况付费的性质可以采取设备租赁的形式，设备租赁保证了云提供商提供一种最低的服务水平。

虚拟化是此模式的一个关键特性。早在几年前，IT 机构就已经明白虚拟化使他们可以方便快捷地创建已有环境的副本——有时涉及多个虚拟机——来支持测试、开发和分级 (Staging) 活动。这些环境的成本极小，因为它们几乎不使用什么资源，因而可以与生产环境共处于同样的服务器之上。

同样地，可以在已有服务器上的新虚拟机中开发和部署新应用程序，在互联网上开放使用，并且在应用程序在市场上取得成功时进行扩展。这种轻便的部署模式已经产生一种“进化式” (Darwinistic) 业务开发方法，其中，软件的 Beta 版是对公众开放的，而且由市场决定哪些应用程序值得进一步扩展和开发，或者静静地报废。

云计算通过自动化扩大了这一趋势。不是与 IT 机构洽谈购买用来部署应用程序的资源，计算云是一个自助式命题，其中，一张信用卡即可购买计算周期，而且可以使用 Web 接口或 API 创建虚拟机，并在虚拟机之间建立网络关系。云不需要与 IT 机构或服务提供商签订长期服务合同，而是按照根据使用情况付费或按 Sip 付费 (Pay-by-the-Sip) 的模式运作，在这种模式下，一个应用程序可能为运行一项作业几分钟或几小时而存在，也可能为长期向客户提供服务而存在。构建计算云时就好像应用程序是临时的，而计费是按照资源消耗情况进行的：使用的 CPU 小时数、移动的数据量或存储的数据的千兆字节 (GB) 数。

使用和仅对使用的资源付费的能力，把购买多少基础设施的风险，从开发应用程序的机构转移给云提供商。这种能力还把架构决策的责任从应用程序架构设计师转移给开发人员。这种转移会增大风险，即出于某种原因制定了流程的企业必须控制的风险，以及系统、网络和存储架构设计师需要把云计算设计包括在内的风险。

基础设施是可以编程的

这种架构责任的转移产生重要的后果。过去，架构设计师确定一个应用程序的各种组件如何在—组服务器上—进行布局，即如何连接、固定、管理和扩展这些组件。现在，开发人员可以使用云提供商的 API 不仅在虚拟机上创建应用程序的初始结构，而且还确定该应用程序如何扩展和演进以适应工作负荷的变化。

看看下面这个类比：历史上，使用 Java 编程语言编写软件的开发人员确定何时适合创建使多项活动同时推进的新线程。现在，开发人员可以同样轻而易举地发现和连接一项服务，使它们可以将一个应用程序扩展到这样一个高度：该应用程序可使用成千上万个虚拟机来适应需求激增情况。

动态编写应用程序架构的程序的能力使开发人员拥有了巨大权力，同时也承担相应大的责任。要最有效地使用云计算，开发人员还必须是架构设计师，而且该架构设计师需要能够创建自我监控和自我扩展的应用程序。该开发人员/架构设计师需要清楚何时适合创建一个新的线程 (而不是何时创建一个新的虚拟机)，并创建如何把它们相互连接起来的架构模式。

一旦很好地理解并利用这种能力，结果将会是蔚为壮观的。一个已经具有传奇色彩的故事是 Animoto 的聚合工具，该工具从一组映像和音乐中创建一个视频。该公司的应用程序在仅仅三天时间里从 50 台服务器扩展到 3500 台服务器，这部分是因为一个使该应用程序能够容易地扩展的架构。为了达到这一目的，该应用程序必须设计为可以水平扩展、具有有限的状态，并且通过云 API 管理自己的部署。对于每个像这样的成功案例，都可能会成为一个相似的故事：其中该应用程序不能自我扩展，而且无法满足消费者的需求。这种从开发人员到开发人员/架构设计师的转移的重要性是无法理解的。

看看您的企业数据中心是否能够以这么快的速度将一个应用程序扩展为适应如此快速增加的工作负荷，以及云计算是否可以测量您的当前能力。

应用程序是组合在一起的，而且设计为可以组合的

这种自助式、按使用情况付费的模式另一个后果是，就像编写应用程序一样，通过汇编和配置设备和开放源软件来组合应用程序。可以重构 (Refactor) 以最大限度地利用标准组件的应用程序和架构，是那些将会在利用云计算效益方面最为成功的应用程序和架构。同样地，应用程序组件应设计为可以组合的，这种组合是通过将应用程序构建的易于使用来实现。这要求具备简单而明确的功能以及精心编写文档的 API。构建大型完整应用程序已成为过去，因为可直接使用或根据特定用途定制的现有工具库已经变得越来越大。

有关如何完成这一壮举的说明，请访问：<http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/01/self-service-prorated-super-computing-fun/>

例如，像 Hadoop (一种开放源 MapReduce 架构实现) 这样的工具可以在多种情况下使用，其中可以对一个问题及其数据进行重构，以便于其多个部分可以同时执行。当《纽约时报》想将其档案中 1100 万份文章和映像转换成为 PDF 格式时，其内部 IT 机构说这会需要七个星期时间。同时，使用 100 个运行 Hadoop 的 Amazon EC2 简单 Web 服务接口实例的一名开发人员，用 24 小时时间就完成了这项工作，劳动成本只有 300 美元。(这不包括上传数据所需的时间或存储成本。)

甚至大型企业都可以使用云计算，用来以比传统企业计算更少的时间和成本解决重大问题。

Web 应用程序部署示例

举一个虚拟化与自助服务结合在一起如何促进应用程序部署的示例，看看如何在云中进行一次二层 Web 应用程序部署 (图)：

1. 开发人员可以从一个预配置虚拟机映像库中选择负载均衡器 (Load Balancer)、Web 服务器和数据库服务器设备。
2. 开发人员配置每个组件以制作一个自定义映像。配置负载均衡器，通过将静态内容上载到存储云来给 Web 服务器填充这些内容，并用站点的动态内容来填充数据库服务器设备。
3. 开发人员把自定义代码层叠在新的架构之中，从而使组件满足特定应用程序要求。

4. 开发人员选择一个呈现各层映像并部署这些映像的模式，以便于处理网络、安全和可扩展性问题。

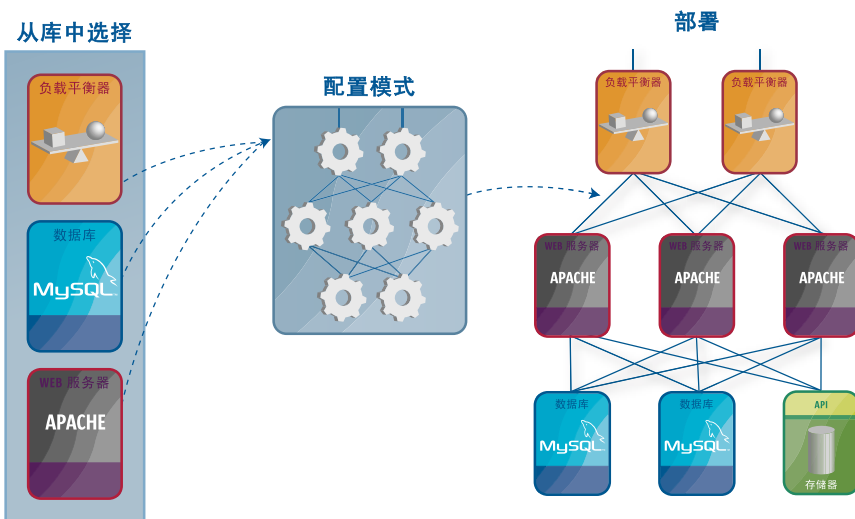


图 1. 以基于云的部署形式将应用程序部署到一个二层 Web 服务器架构模式的示例。

5. 安全而高度可用的 Web 应用程序启动并正常运行。当需要更新应用程序时，可以在开发-测试-生产链之间对虚拟机映像进行更新、版本控制和复制，并且可以重新部署整个基础设施。云计算假定一切都是临时的，而且重新部署整个应用程序就像手动修补一组具体的虚拟机一样容易。

在此示例中，虚拟机映像的抽象性质支持一种组合式应用程序开发方法。通过重构该问题，可以使用一组标准的组件来快速部署应用程序。借助此模式，可以迅速满足企业的业务需要，而不需要对服务器、存储和网络基础设施进行耗时的人工购买、安装、布线和配置工作。

通过网络提供服务

不言而喻，云计算扩大了通过网络提供服务的已有趋势。几乎每个商业机构都认可与其应用程序的连接的基于 Web 的接口的价值，不管是否通过互联网向客户提供应用程序，也不管它们是否是可供授权员工、合作伙伴、供应商和咨询人员使用的内部应用程序。当然，基于互联网的服务提供的美妙之处就在于可以随时随地使用应用程序。

尽管企业都清醒地认识到利用安全套接字层 (SSL) 以及严格验证技术来确保通信安全的能力，但在云计算环境中引入信任还需要认真考虑企业计算和云计算之间的区别。如果架构设计的合理，互联网服务提供模式可提供各种规模的企业所需的灵活性和安全性。

开放源软件的作用

开放源软件在云计算中发挥着一种重要的作用，因为开放源软件允许从容易访问的组件创建其基本软件元素：虚拟机映像和设备。这会产生巨大的影响：

- 例如，开发人员可以通过将 MySQL 软件层叠在一个 OpenSolaris™ 操作系统上并执行自定义来创建一个数据库设备 (图 2)。像这样的设备能够根据需要创建、部署和动态扩展云计算应用程序。例如，看看开放源软件如何使 Animoto 创建的应用程序在几天之内就扩展到 3500 个实例。



图 2. 可以通过把开源软件层叠在一个虚拟机映像之中，并执行简化其部署的自定义，来创建设备。在此示例中，通过把 MySQL 软件层叠在 OpenSolaris 操作系统上来创建一个数据库设备。

- 由于用开放源组件汇编大型应用程序非常容易，因而生成更多开放源组件。这反过来又使开放源软件的作用更加重要。例如，需要拥有一种可在云计算环境中运行的 MapReduce 算法，这就是刺激开发该算法的因素之一。既然创建了工具，就可以用它来进一步提高开发人员编写云计算应用程序的水平。

云计算基础设施模式

当从一个标准企业应用程序部署模式向一个基于云计算的应用程序部署模式转变时，云计算架构设计师需要考虑许多问题。有的公用云和专用云提供互补的优点，有三种基本服务模式需要考虑，并且需要对比开放 API 和专有 API 的价值。

公用云、专用云和混合云

IT 机构可以选择在各有其取舍的公用云、专用云或混合上部署其应用程序。公用、专用与混合这几个术语并不规定位置。公用云一般就在互联网上，而专用云通常在建筑物内，还有可能设在主机托管场所。

企业可以就选用哪种云计算模式考虑多种因素，而且有可能选用不只一种模式来解决多种不同问题。如果是临时需要的应用程序，可能最适合在公用云上部署，因为这样可以避免为了临时的需要而购买额外设备的情况。同样地，永久使用或对服务质量或数据位置有具体要求的应用程序，最好在专用云或混合云上部署。

公用云

公用云由第三方运行，而不同客户提供的应用程序可能会在云的服务器、存储系统和网络上混合在一起 (图 3)。公用云通常在远离客户建筑物的地方托管，而且它们通过提供一种像企业基础设施进行的灵活甚至临时的扩展，提供一种降低客户风险和成本的方法。

如果在实施一个专用云时牢记性能、安全性和数据保存位置，那么，该云中运行的其它应用程序的存在应对云架构设计师和最终用户都是透明的。的确，公用云的优点之一是，它们可以比一个公司的专用云大很多，因而能够根据需要进行伸缩，并将基础设施风险从企业转移到云提供商——哪怕仅仅是临时性的。

可以将公用云的部分划出去，以便于独占单个客户端，从而产生一个虚拟专用数据中心。虚拟专用数据中心不是仅限于在公用云中部署虚拟机映像，而是使客户在更大程度上清楚地了解其基础设施。现在，客户不仅可以处理虚拟机映像，而且可以处理服务器、存储系统、网络设备和网络拓扑。利用位于同一场所的所有组件创建一个虚拟专用数据中心，有助于缓解数据位置问题，因为当在同一场所内连接资源时，带宽非常充足，而且一般都可用。

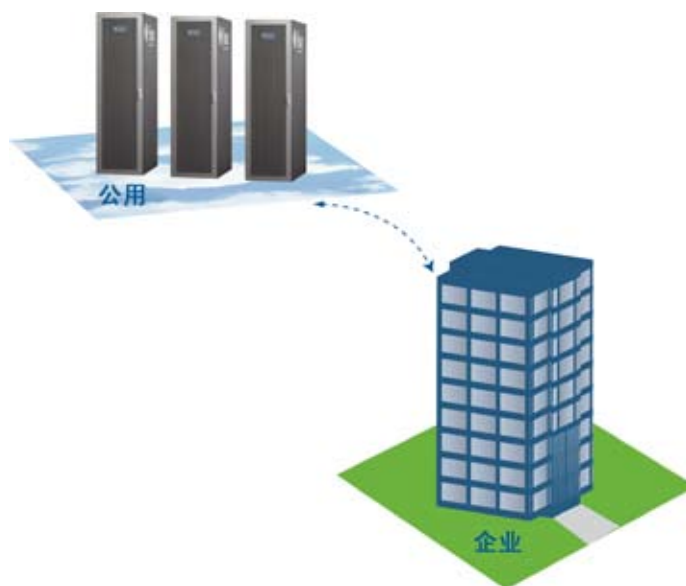


图 3. 公用云向多个客户提供服务，而且，一般在主机托管场所进行部署。

专用云

专用云是为一个客户单独使用而构建的，因而提供对数据、安全性和服务质量的最有效控制 (图 4)。该公司拥有基础设施，并可以控制在此基础设施上部署应用程序的方式。专用云可部署在企业数据中心中，也可以将它们部署在一个主机托管场所。

专用云可由公司自己的 IT 机构也可由云提供商进行构建。在此“托管式专用”模式中，像 Sun 这样的一个公司可以安装、配置和运营基础设施，以支持一个公司企业数据中心内的专用云。此模式赋予公司对于云资源使用情况的极高水平的控制能力，同时带来建立并运作该环境所需的专门知识。

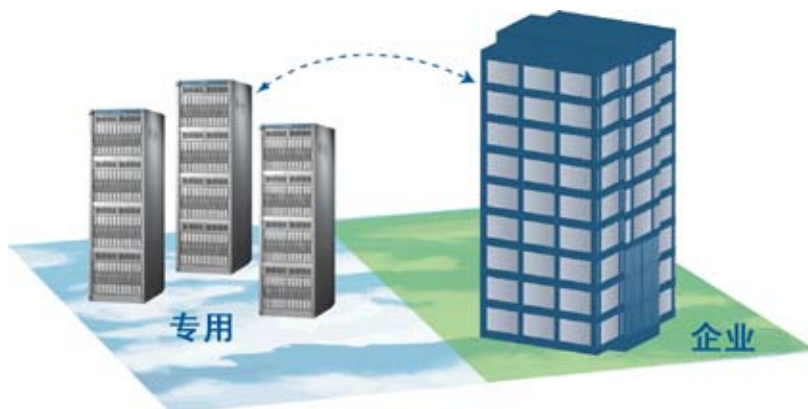


图 4. 专用云可以托管在某个主机托管场所，也可以托管在企业数据中心之中。它们可以由公司支持，也可以由云提供商提供支持，还可以由像外包公司这样的第三方提供支持。

混合云

混合云把公用云模式与专用云模式结合在一起 (图 5)。混合云有助于提供按需的、外部供应的扩展。用公用云的资源扩充专用云的能力可用在发生工作负荷快速波动时维持服务水平。在利用存储云支持 Web 2.0 应用程序时，这最常见。混合云也可用来处理预期的工作负荷高峰。专用云，有时称为“超负荷计算” (Surge Computing)，可用来执行易于在公用云上部署的定期任务。

混合云引出确定如何在公用云与专用云之间分配应用程序的复杂性。需要考虑的问题包括数据和处理资源之间的关系。如果数据量小，或应用程序无状态，与必须把大量数据传输到一个公用云中进行小量处理相比，混合云要成功得多。

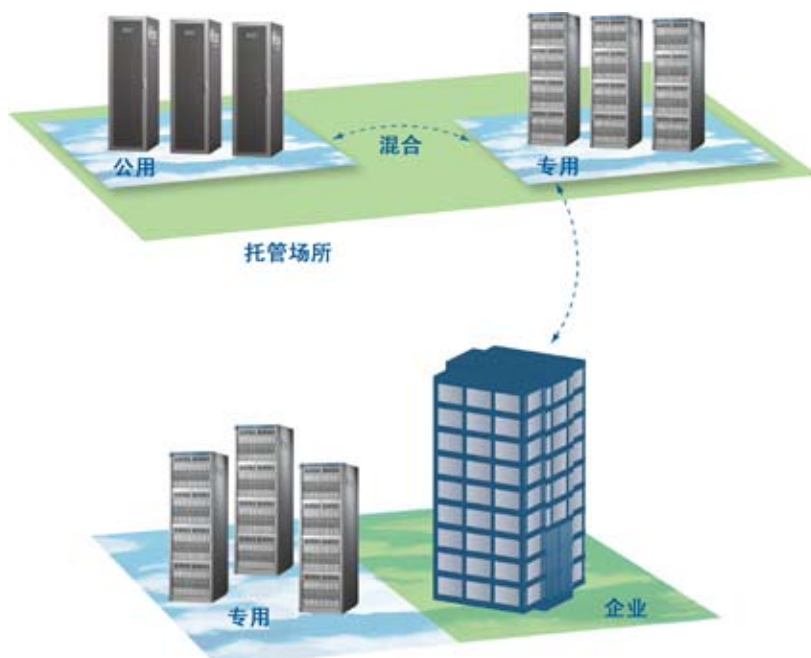


图 5. 混合云把公用云模式和专用云模式结合起来，而且，当这两种类型的云位于同一场所时，混合云特别有效。

云计算的架构层

Sun 公司的云计算观点是一种全面的观点：云计算可描述在从硬件到应用程序的任何传统层级提供的服务 (图 6)。实际上，云服务提供商倾向于提供可分为如下三个类别的服务：把软件当作服务 (Software as a Service)、把平台当作服务 (Platform as a Service) 以及把基础设施当作服务 (Infrastructure as a Service)。这些类别把如图 6 所示的各种层级组合在一起，其中存在某种重叠。

把软件当作服务 (SaaS)

“把软件当作服务”的特色是根据需要作为服务提供的一整套应用程序。该软件的单个实例运行于云上，并为多个最终用户或客户机构提供服务。

最著名的 SaaS 示例是 salesforce.com，不过许多其它实例已经进入市场，其中包括提供基本商业服务 (包括电子邮件和文字处理) 的 Google Apps。

尽管 salesforce.com 诞生比云计算定义的出现早了几年，但它现在通过利用其伴侣 force.com 运作，后者可定义为“把平台当作服务”。

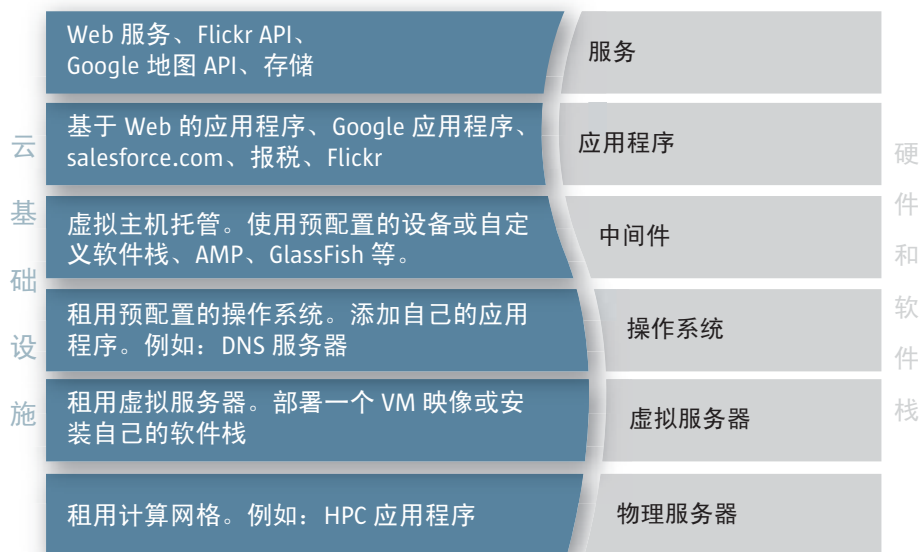


图 6. 云计算意味着把 IT 基础设施用作一项服务，而且该服务可以是租用原始硬件到使用第三方 API 的任何事情。

把平台当作服务 (PaaS)

“把平台当作服务”包含一个软件层，并作为一项服务提供此软件层，这项服务可用来构建更高水平的服务。从服务生产商或消费者的观点看，关于 PaaS 的观点至少有两种：

- 生产 PaaS 的某个人可能通过集成 OS、中间件、应用软件甚至是一个随后作为服务提供给客户的开发环境来生产一个平台。例如，开发 PaaS 的人可能会使其以一组 Sun™ xVM 管理程序虚拟机为基础，这组虚拟机包括一个 NetBeans™ 集成化开发环境、一个 Sun GlassFish™ Web 栈，并支持像 Perl 或 Ruby 这样的其它编程语言。
- 使用 PaaS 的人会看到一个封装式服务，该服务是通过 API 提供给用户的。客户通过 API 与该平台互动，而且该平台执行一切必要的操作来管理和扩展其本身，以提供规定的服务水平。虚拟设备可以归类为 PaaS 的实例。例如，一个内容交换设备会将其所有组成软件对客户隐藏起来，而只向客户提供用来配置和部署服务的一个 API 或 GUI。

PaaS 产品可执行各个阶段的软件开发和测试，也可以专用于某个领域，例如，内容管理。PaaS 的商业示例包括 Google App Engine，它在 Google 的基础设施上提供应用程序服务。像这样的 PaaS 服务可提供一个用来部署应用程序的强大基础，然而它们可能会由于云提供商选择提供的能力而受到制约。

把基础设施当作服务 (IaaS)

“把基础设施当作服务”通过网络作为标准化服务提供基本存储和计算能力。服务器、存储系统、交换机、路由器和其它系统都是合用的，并可用来处理从应用程序组件到高性能计算应用程序的工作负荷。

IaaS 的商业示例包括 Joyent，其主要产品是提供高度可用的按需基础设施的一系列虚拟化服务器。

云应用程序设计接口

区分云计算与标准企业计算的关键特征之一是：基础设施本身是可以编成程序的。开发人员不是实际部署服务器、存储器和网络资源来支持应用程序，而是指定如何配置和互连同样的虚拟组件，包括如何从存储云存储和检索虚拟机映像和应用程序数据。开发人员指定如何且何时通过云提供商指定的 API 来部署组件。

这与文件传输协议 (FTP) 的工作方式相似：FTP 服务器维持与会话期间一直开放的客户端的控制连接。当要传输文件时，该控制连接用来向服务器提供一个来源或目标文件名，并协商一个用于文件传输本身的源和目标端口。从某种意义上讲，云计算 API 就像 FTP 控制信道一样：云计算 API 在使用云期间是开放的，而且控制如何利用云来提供开发人员向往的最终服务。

控制如何利用云基础设施的 API 的使用具有一个缺陷：与 FTP 协议不同的是，云 API 尚未标准化，因此，每个云提供商都有自己用来管理其服务的特定 API。这是一个新兴行业的典型状态，其中，每个供应商都有其专有技术，这样的技术往往把客户限制在其服务里，因为专有 API 使得变更提供商非常困难。

请查找在尽可能多的地方使用标准 API 的提供商。标准 API 现在可用来访问存储设备，而随着时间的推移，用来部署和扩展应用程序的 API 可能会被标准化。另外，请查找有哪些云提供商了解其市场而且提供存档并部署虚拟机映像和预配置设备的方法。

云计算效益

要想从云计算中实现最大效益，开发人员必须能够重构其应用程序，使应用程序可以最有效地利用云计算所支持的架构和部署方式。使用云计算部署应用程序的优点包括缩短运行时间和响应时间、最大限度地减轻部署物理基础设施的风险、降低入市成本以及加快创新步伐。

缩短运行时间和响应时间

对于弹性地使用云来运行批量作业的应用程序来说，云计算使得使用 1000 台服务器在相当于单个服务器所需的千分之一的时间里完成一项任务变得非常简单。前面引用的《纽约时报》的示例就是一个很好的批作业示例，通过利用云，极大地缩短了其运行时间。

对于需要向其客户提供良好响应时间的应用程序来说，重构应用程序以便把任何 CPU 密集型任务外包给“工人”(Worker) 虚拟机，有助于优化响应时间，同时还能根据需求进行伸缩，从而满足客户需求。前面引用的 Animoto 应用程序就是云如何用来扩展应用程序并维持服务质量水平的一个很好的示例。

最大限度地减轻基础设施风险

IT 机构可以利用云来减轻购置物理服务器所固有的风险。新的应用程序是否会成功？如果成功，需要多少台服务器？部署这些服务器的步骤是否能够跟得上工作负荷增加速度？如果不能，投入服务器中的大量资金会不会付之东流？如果该应用程序的成功非常短命，IT 机构是否还会在多数时间里处于空闲状态的大量基础设施中进行投资？

当把一个应用程序推出到云中时，可扩展性和购买太多或太少基础设施就成为云提供商的问题。越来越多的案例表明，云提供商的基础设施规模如此之大，以至于可以容忍各个客户的业务量增长和工作负荷尖峰情况，因而减轻了这些客户所面临的经济风险。

云计算最大限度地减轻基础设施风险的另一条途径是实现超负荷计算，其中企业数据中心(也许是实现专用云的数据中心)通过一个允许其向一个公用云发送超溢工作来扩大其处理工作负荷尖峰情况的能力。在一个资源不再稀缺而且能够以较低成本更好地满足资源需求的环境中，可以更好地处理应用程序生命周期管理问题。

降低入市成本

云计算的许多属性有助于降低进入新市场的成本。

- 由于基础设施是租用的，而不是购买的，成本得到控制，而且资本投资可能为零。除通过按 Sip 购买计算周期和存储空间来降低购置成本之外，云提供商的巨大规模也有助于最大限度地降低成本，从而有助于进一步降低入市成本。
- 应用程序与其说是通过汇编倒不如说是通过编程来开发的。这种快速应用程序开发方法非常规范，有助于缩短入市时间，因而有可能使在云环境中部署应用程序的机构先于竞争者入市。

加快创新步伐

云计算有助于加快创新步伐。降低进入新兴市场的成本有助于使竞争各方处于同一起跑线，因而使新创企业可以快速而低成本地部署新的产品。这使小公司可以更有效地与在企业数据中心领域里所经历的部署过程长得多的传统机构进行竞争。增强竞争能力有助于加快创新步伐，而且由于许多创新是通过利用开放源软件实现的，整个行业都会从云计算技术所促成的创新步伐加快而受益。

第 3 章

实现 IaaS 必须考虑的架构问题

不断发展的应用程序架构

正如我们已经说明的：云计算是当前趋势和最佳做法的自然延伸，从架构观点看云计算也是如此。再次强调，云计算并非新生事物，不过在其实现中，云计算改变我们所做的一切工作。

变革架构的途径

在 20 世纪 90 年代，人们谈论的话题是关于如何把一个应用程序分解成各种组件，然后是如何将这些组件部署到不同的服务器上，以便优化非功能要求，包括可扩展性、可用性、易管理性和安全性。现在，我们所维持的是一个分解式应用程序架构，同时实际部署到一个利用虚拟化的整合式架构上。

云计算通过提供一种有计划地部署应用程序架构的方法来继续这一趋势，最终实现一个动态数据中心的承诺。在云计算中，效率受到高度重视；如果无法快速而有计划地部署应用程序架构，那么该应用程序就可能不是一个适合于此种模式的应用程序。

变革应用程序设计

过去，应用程序设计为通过垂直扩展来处理大型工作负载。将更多处理器和内存安装在一个邮件服务器上处理更大的流量。扩展数据中心服务器以提高吞吐量。在超级计算机上运行高性能计算作业。

从高度可扩展的对称型多处理器迁移到价格低廉但可扩展性差的 x86 架构服务器已经对应用程序设计产生了影响。开发人员并不期望应用程序运行于高度可扩展的服务器，而是重构其应用程序，以便能够在多个服务器之间进行水平扩展。此应用程序重构并不总是容易进行的，因为应用程序及其数据都必须进行设计，这样才能把处理和数据同时分解成为较小的数据块。这一已有架构趋势已经成为推广云计算的一个关键因素。这一趋势的示例包括：

高性能计算

高性能计算 (HPC) 工作负荷已经在光金属 (Bare-Metal) 计算网络上运行有一段时间了，这是通过应用程序重构实现的。例如，科学家已经找到为像 3D 气候建模这样的应用程序削减数据的方法，因而可以在许多服务器上展开应用程序。网格计算 (Grid Computing) 是云计算的“前辈”，这是因为网格计算使用工具来供应和管理多架物理服务器，这样，这些服务器就可以共同配合来解决一个问题。由于具有极高的计算、互处理通信和 I/O 需求，HPC 工作负荷非常适合于作为服务提供基础设施的云，尤其是提供对 I/O 设备进行更直接访问的光金属服务器或 Type I 虚拟机。

数据库管理系统

数据库管理系统通过水平扩展数据库服务器并在其之间对表进行分区，已经适应于在云环境中运行。此技术 [称为“分片” (Sharding)] 使多个数据库软件实例 (通常是 MySQL 软件) 可以在云环境中扩展性能。应用程序现在不是访问单个中央数据库，而是访问多个数据库实例中的一个，具体取决于哪个“碎片” (Shard) 包含所需数据 (图 7)。



图 7. 数据库分片技术在多个数据库管理系统实例之间对数据库表进行分区，可通过水平扩展来支持大型数据库。

CPU 密集型处理

执行像帧渲染 (Frame Rendering) 这样的活动的应用程序已经设计为不是为每个帧创建一个新线程，而是创建一个单独的虚拟机来渲染每个帧，因而通过水平扩展来提高性能。

数据密集型处理

开放源社区正在开发一般化工具，这些工具可帮助处理大量数据，然后将结果聚集到相应的进程中。例如，Hadoop 就是 MapReduce 问题的一种开放源实现，这种实现将“工人” (Worker) 虚拟机与他们所需的数据的部署集合在一起。

目标仍然相同

应用程序架构方面的许多进步都有助于促进云计算的推广。这些进步有助于支持高效地部署应用程序的目标，同时有助于使应用程序具有弹性，并适度地自动扩展。然而，杰出应用程序架构的首要目标始终没有发生任何变化：那就是支持同样的始终至关重要的特征：

- **可扩展性。** 此特征的重要性始终未曾改变。为云计算设计的应用程序需要根据工作负荷需求进行伸缩，这样，性能及与服务水平的符合性就保持适当。为了达到这一点，必须将应用程序及其数据松散耦合，以使可扩展性最大

化。“弹性”这一术语经常适用于伸缩云应用程序，因为云计算应用程序必须不仅可以扩展，而且必须在工作负荷减小时能够缩小，以免抬高在云中的部署成本。

- **可用性。**无论应用程序是向社交网站用户提供服务，还是管理大型制造企业的供应链，互联网应用程序用户都期望这些应用程序每时每刻都能正常运行。Sun 公司在此领域的行业领先地位早在其推出 SunToneSM 认证计划时就已确立，该计划有助于客户证明其应用程序和服务达到要求的可用性水平。
- **可靠性。**随着时间的推移，对于可靠性的重视程度已经发生转变。如果大型应用程序是指大型对称多处理系统，那么可靠性就意味着系统组件极少发生故障，而且即使发生故障，也可以在不中断系统正常运行的情况下进行更换。现在架构适应此特征的方式是把应用程序设计为：即使分解运行应用程序的一个或多个服务器或虚拟机，应用程序仍能继续运行，而且其数据不会受到破坏。正是在这个方面，我们曾经担心个别服务器组件发生故障，而现在我们构建即使全部服务器发生故障也不会造成破坏的应用程序。
- **安全性。**设备需要只对获得授权且通过身份验证的用户提供访问权，而且这些用户需要能够确信其数据是安全的。无论应用程序帮助各个互联网用户准备其纳税申报表，还是应用程序在公司及其供应商之间交换保密信息，都是如此。现今环境中的安全性是通过如下手段确立的：利用严格的身份验证、授权和帐号管理程序，确保静止和移动中数据的安全性，锁定网络，以及硬化操作系统、中间件和应用软件。安全性是这样一个系统特性，以至于我们不再单独强调安全性 — 必须将安全性融合到应用程序的各个方面以及应用程序部署和操作架构和进程中。
- **灵活性和敏捷性。**这些特征越来越重要，因为商业机构发现他们必须通过提高向客户提供应用程序的速度来更快地适应不断变化的经营环境。云计算强调通过利用最适合的构件 (Building Block) 来快速完成工作，使应用程序非常快速地面市。
- **可维护性。**一旦部署一个应用程序，就需要对其进行维护。过去，可维护性意味着使用的服务器可以在不停机或极少停机的情况下进行修理。现在，可维护性是指，更新甚至更换一个应用程序的基本基础设施组件时，不会破坏该应用程序的特征，其中包括可用性和安全性。
- **效率。**这是特征列表上的新特征，或许这是最能将云计算模式与其它计算模式区别开来的一个特征。效率是云计算的意义所在，而且如果不能方便快捷地在云中部署应用程序，即使可从模式中受益，该计算模式也可能不是一个良好的可选模式。例如，企业资源规划应用程序可能最适合于垂直扩展的系统，并可能在近期通过 SaaS 提供。不过，提取、处理和呈现这些系统中派生的数据的应用程序可能非常适合于在云中部署。

一致而稳定的抽象层

云计算提高了抽象水平，这样，所有组件都抽象化或虚拟化，并可用来迅速组合较高级别的应用程序或平台。如果某个组件不向其客户或同行提供一致而稳定的抽象层，该组件就不适合于云计算。

标准部署单位是虚拟机，它本质上可运行于抽象硬件平台。人们很容易过度关注构建虚拟机映像，而忽视用来创建虚拟机映像的模式。在云计算中，维持该模式而非映像本身非常重要。该模式是保留下来的，而映像则是从该模式产生的。

虚拟机映像将始终在变化，因为虚拟机映像内的软件层将总是需要修补、升级或重新配置。不变的是创建虚拟机映像的流程，而且这是开发人员所应重视的。开发人员可以通过把 Web 服务器、应用程序服务器和 MySQL 数据库服务器层叠在一个操作系统映像上，应用补丁程序、配置更改，以及互连各层组件，来构建虚拟机映像。重视模式，而非虚拟机映像，可以通过重新把模式应用到一组新组件，根据需要来更新这些映像本身。

凭借这一标准部署单位，云架构设计师可以使用有助于以较低成本加快部署速度的设备。开发人员可以使用一个设备，该设备预配置为通过与该设备的 API 进行互动，在 OpenSolaris OS 上运行 Hadoop。架构设计师可以使用内容交换机，这些内容交换机不是作为物理设备部署的，而是作为虚拟设备部署的。部署该设备所需要做的一切事情只是与其 API 或 GUI 进行互动。即使生产带有许可证的商用软件的公司都在通过更加灵活、基于使用情况的许可模式来适应云计算。

无论是调用一个创建虚拟机映像的模式，还是定制一个设备，结果产生的虚拟机映像都需要存储在企业进行版本控制并提供支持的映像库中。

标准有助于解决复杂问题

云计算首先重视效率，因而采用少数标准和标准配置有助于降低维护和部署成本。拥有可简化部署的标准比拥有用于作业的最佳环境更重要。80/20 规则就在这里发挥作用：云计算重视可以支持 80% 使用案例的少数标准。这就把经济情况从成本高的一次性实现转变为选择可最大限度地加以利用的构件。将来还会继续专业化，但起点应从标准开始。

对于要采用云计算的企业，标准可以包括虚拟机类型、标准虚拟机映像中的操作系统、工具以及支持的编程语言。

- **虚拟机类型。**想想虚拟机选择对于要支持的应用程序的影响。对于社交网站应用程序、出于安全性进行的隔离，以及出于可移植性进行的高水平抽象，会建议使用 Type II 虚拟机。对于高性能计算或可视化应用程序，需要直接访问硬件以实现最佳性能，会建议使用 Type I 虚拟机。

- **预安装、预配置的系统。**必须像在物理服务器上一样维护虚拟机上的软件。操作系统仍然需要硬化、修补和升级。拥有一小组标准化的受支持配置，使开发人员可以使用当前支持的虚拟机。当升级支持的配置时，应设计要求自定义的模式，以便于容易地将更改重新应用到一个新的虚拟机映像。设备也是如此，其中，可以通过设备的标准 API 来配置当前版本。
- **工具和语言。**企业可能以标准方式使用 Java 编程语言和 Ruby on Rails；小企业可以以标准方式将 PHP 作为其用于构建应用程序的首选工具。当这些标准在云计算上下文中成熟时，它们开始形成下一层：把平台当作服务 (PaaS)。

虚拟化和封装技术支持重构

当通过组合和配置一组虚拟机映像和设备重构并创建应用程序时，要将重点放在特定虚拟机发挥什么作用上，而不是放在如何实现该虚拟机上。虚拟化和封装技术将实现细节隐藏起来，并使开发人员重新重视组件之间的接口和互动。这些组件应该提供标准接口，以便于开发人员方便快捷地构建应用程序，同时利用与性能或成本所要求的相似的功能来使用替代组件。

应用程序开发是有计划地完成的，甚至用来部署应用程序的程序也可以封装，以便于利用和重新利用。可以封装部署三层式 Web 基础设施的程序，这样，该程序的参数就会包括指向用于 Web 服务器、业务逻辑和数据库层的虚拟机映像的指针。然后就可以执行此设计模式，以便于部署标准应用程序，而不必重新设想或甚至重新考虑 (例如) 支持每层所要求的网络架构。

应用程序维护的云计算原则并非修补，而是重新部署。管理创建虚拟机映像的模式，而不是映像本身，简化了这种重新部署。部署之后发现的问题解决起来相当容易，或者通过更新组件虚拟机并调用用于重新部署的设计模式，来发布应用程序的新版本。当开发人员修补虚拟机时，只需要创建一个虚拟机映像，而且要有计划地复制并部署其余映像。应对虚拟机进行版本控制，以便于必要时回滚。

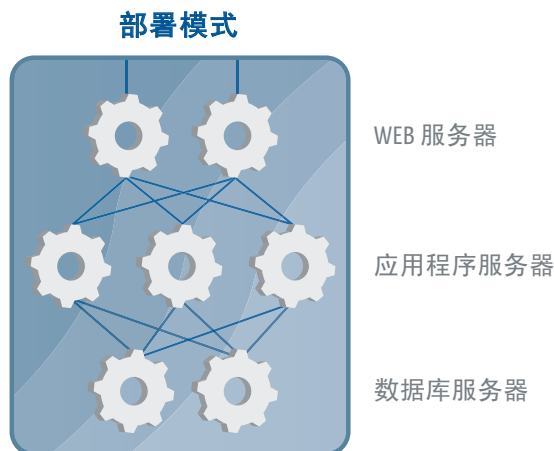


图 8. 一种部署模式可以封装起来，以便于重复利用。在此示例中，一个模式指定 Web 服务器、应用程序服务器和数据库服务器层，而用此模式部署其本身的一个实例所需要的一切只是指向上述三层中各层的虚拟机的指针。

松散耦合、无状态、原地失败 (Fail-in-Place) 计算

几年以来，基于 Web 的应用程序已在向松散耦合和无状态转变。在云计算中，这些特征甚至更加重要，因为云计算具有更加动态的性质。应用程序映像不修补，它们是用完丢弃的对象，因而需要是无状态的。如果一个虚拟机失败，应用程序必须继续不间断地运行。应用程序之间的耦合需要是松散的，这样，任何组件发生故障都不会影响整个应用程序的可用性。一个组件应该做到“原地故障”，极少甚至不会对应用程序产生影响。

由于应用程序组件越来越具有临时性，因而不能包含存在时间超过任何应用程序实例的数据。应通过将状态推出软件来尽可能使应用程序无状态，从而尽可能地将处理与数据分开。能够做到这一点的技术包括：

- 以 Cookie 的形式将状态推出到用户，或者将状态代码编入 URL。
- 将状态下推至后端数据库
- 维护数据的补充副本，这是 Hadoop 使用的一个策略
- 使用基于网络的持久性技术，例如，GlassFish 应用程序服务器中的 Terracotta 或 Shoal。

“原地失败”计算对操作活动的影响是，即使是硬件也应该是无状态的，以便于云正常工作。硬件配置应存储在元数据中，这样，在发生故障时就能够恢复配置。

水平扩展

云计算使得大规模水平可扩展性可以用于有条件利用的应用程序。现在出现一个设计并重构应用程序以在水平扩展环境中顺利工作的趋势，该趋势意味着越来越多的应用程序能够很好地适应云计算。

利用水平扩展的应用程序应该重视假如个别组件发生故障整个应用程序的可用性。多数云平台是在一个虚拟服务器资源池上构建的，其中，如果任何一个物理服务器发生故障，该服务器托管的虚拟机只是在一个不同的物理服务器上进行重构。把无状态和松散耦合的应用程序组件与水平扩展技术结合在一起，可促成一种“原地失败”策略，这种策略与任何一个组件的可靠性都没有关系。

水平扩展不必局限于单个云。根据应用程序数据的大小和位置，“超负荷”计算可用于扩展一个云的能力，以适应临时性工作负荷增加。在超负荷计算中，运行于专用云的应用程序可能会根据需要吸纳来自公用云的额外资源。

Project Kenai 生产的 OpenSolaris 动态服务容器 (OpenSolaris Dynamic Service Containers) 提供一个轻便的供应系统，可用来水平扩展 Solaris 区域 (Solaris Zones)。请访问 <http://kenai.com/projects/dsc/>

超负荷计算很大程度上取决于数据的数量和位置。如果是一个位于企业数据中心的专用云，而且该企业数据中心扩展为使用互联网上某个其它位置的一个公用云，需要移动到公用云的数据的数量需要分解为等式 (参阅下面的“数据物理”一节)。如果是一个驻留在与公用云提供商相同的托管场所的专用云，数据位置问题就会大大减小，因为几乎无限的自由带宽可以连接这两个云。

并行化

水平扩展和并行化如影随形，不过，现在扩展和实现发生了变化。从微观上讲，软件可以在对称式多处理器上使用垂直扩展来产生多个线程，其中，并行化可加快操作速度或改进响应时间。但是，随着现今的计算环境转变为包含两个和四个插槽的 x86 架构服务器，垂直扩展只是像服务器拥有核心的数量一样多地拥有并行处理能力 (或者购买一样多的核心并分配给特定虚拟机)。从宏观上讲，可以在多个服务器之间使用并行化的软件也可以扩到成千上万个服务器，从而提供比用对称式多处理技术实现的更大可扩展性。

在实体世界中，并行化技术通常是通过在多个服务器之间分配入站请求的负载均衡器或内容交换机实现的。而在云计算世界里，并行化可以通过在多个虚拟机之间分配入站请求的负载均衡设备或内容交换机来实现。无论是上述哪种情况，应用程序都可设计为吸收补充资源来适应工作负荷尖峰。

通过负载均衡实现并行化的经典示例是：许多同时访问相同数据的无状态 Web 服务器，其中，在服务器池内分配入站工作负荷 (图 9)。

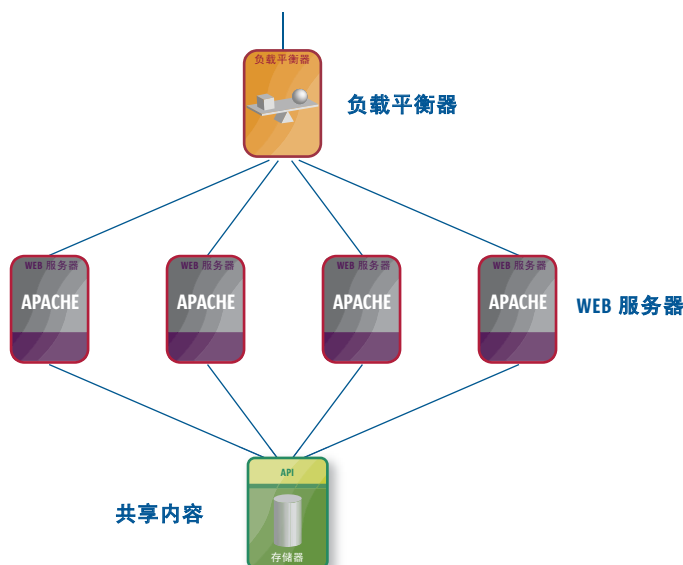


图 9. 并行化与负载均衡的一个十分常见的使用情况是水平扩展式 Web 服务器。

在云计算环境中使用并行化还有许多其它途径。使用巨量 CPU 时间处理用户数据的应用程序可能会使用如图 10 中所示的模式。一个调度程序接收来自用户的作业，将数据存入存储库，然后针对每项作业启动一个新的虚拟机，交给该

虚拟机一个令牌，此令牌允许该虚拟机从存储库中检索数据。当虚拟机完成其任务时，它将令牌传回调度程序，调度程序允许虚拟机将完成的项目返回给用户，并且虚拟机终止工作。

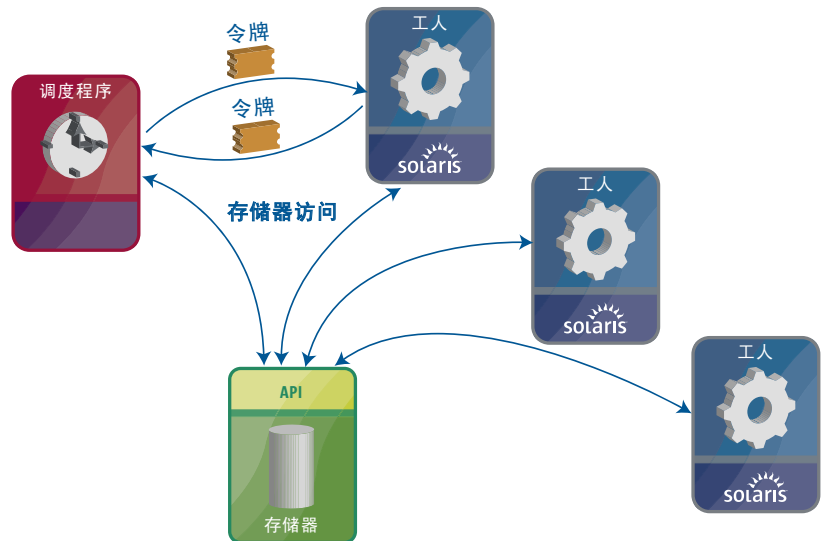


图 10. 并行化的另一个示例是代表用户成批执行资源密集型任务。

分割并征服

应用程序只能并行化到可以对其数据进行分区的地步，这样，独立系统就可以在其上面并行工作。一个好的应用程序架构包括一个分割并征服数据的计划，而其多种真实示例阐释了多种多样的方法：

- Hadoop 是 MapReduce 模式的一种实现，该模式又是主/工人并行化模式的一种实现。
- 数据库分片可通过多种分区技术来完成，其中包括垂直分区、基于范围的分区或基于目录的分区。使用的方法完全取决于将要如何使用数据。
- 大型金融机构已经重构其欺诈检测算法，这样，曾经是大部分批数据挖掘操作的程序现在并行运行于大量系统之上，对输入数据进行实时分析。
- 现在已经设计出一些处理三维数据的高性能计算应用程序，因而通过一个进程就可以为时间 t 计算出一立方容积 (的气体、液体或固体) 的状态。然后一立方的状态就传递给代表八相邻立方的进程，并为时间 $t+1$ 计算出状态。

数据分区对于通过网络传输的数据的量影响很大，这使数据物理成为下一个考虑事项。

数据物理

数据物理考虑处理元素与处理元素所操作的数据之间的关系。由于多数计算云在云中存储数据，而不是在物理服务器的本地磁盘上存储数据，因而需要花费时间把数据迁移到要处理的服务器上。数据物理由一个简单的公式进行确定，该公式描述在生成、存储、处理和存档数据的位置之间迁移一定量的数据花费多长时间。云擅长于存储数据，并不一定擅长于对数据进行存档并按照预定计划毁坏数据。大量数据，或低带宽管道，延长移动数据所花费的时间：

$$\text{时间} = \frac{\text{字节数} * 8}{\text{带宽}}$$

此公式对于 Moment-by-Moment 式数据处理和长期规划都意义重大。例如，它有助于确定实施超负荷计算策略是否有意义，因为其中把数据移动到公用云可能要比直接处理数据花费更长时间。它还有助于确定将操作从一个云提供商移动到另一个云提供商的成本。无论数据在一个云提供商的数据中心里积聚成什么程度，都必须将其移动到另一个云提供商那里，而这一过程可能花费时间。

移动数据的成本可以用时间和带宽费用来表示。图 5 所示的混合模式 (其中一个公司的专用云托管在其云提供商的公用云那里) 有助于极大地降低成本。托管场所的带宽一般都是充足而空闲的，这使这一策略无论在到处移动数据花费的时间还是费用方面都是一个双赢的命题。

数据与处理之间的关系

数据物理提醒人们考虑数据与处理之间的关系，以及从存储设备将数据移动到处理设备会既花费时间又花费资金。这种关系中需要考虑的一些方面包括：

- 在附近没有计算能力的情况下存储的数据价值有限，而且云提供商应该在这两个组件的网络关系方面保持透明。其管道的大小如何？时间延迟情况如何？连接的可靠性如何？云提供商应提前回答上述问题。
- 云架构设计师应该能够指定虚拟组件和服务的位置，这样，虚拟机与其访问的存储设备之间有一个明确定义的关系。
- 云提供商可以自动为客户优化这种关系，但要考虑其机构是否适合于手头的应用程序。
- 在一个联网环境中，有时计算一个值比从联网存储设备汇总检索它效率更高 (更快，时间延迟更短)。想想使用计算周期与到处移动数据之间的利弊得失。

编程策略

计算云要求使用考虑数据移动的编程策略：

- 到处移动指针通常要比移动实际数据要方便得多。请注意图 10 所示的调度程序/工人模式使用一种中央存储服务，并在应用程序之间传递令牌，而不是传送实际数据。
- 指针应视为一种能力，认真确保指针不容易被伪造。

- 像代表性状态传输 (REST) 或简单对象访问协议 (SOAP) 这样的工具有助于降低应用程序状态，同时还能管理状态数据的传输。

合规与数据物理

保持符合政府法规和行业要求给数据管理增加了另一层需要考虑的因素。云架构设计师需要能够给数据存储指定拓扑和地理约束。云提供商应该设法简化指定数据与处理数据的虚拟机之间的关系以及数据实际存储位置。

- 处理个人数据的公司可能需要遵守与数据处理相关的政府法规。例如，在欧盟国家开展业务的公司如果将其数据存储在美国，那就违反了本地法律，因为两地法律保护个人数据的规定存在差异。在类似情况下，云提供商应该提供一种对数据移动方式和位置指定约束条件的能力。
- 受行业标准制约的公司 (例如，通过处理授权的信用卡征税的行业标准) 可能面临在哪里存储数据以及如何和何时毁坏数据方面的限制。在类似情况下，不允许把空闲磁盘存储块与另一个客户的存储块混合起来。重新使用之前必须安全地擦除这些存储块。

选择云提供商存储数据时，要考虑的不仅仅是该提供商是否可信。还需要考虑云提供商是否按照适用于相应应用程序的标准通过认证。

安全性与数据物理

数据通常是一个公司最有价值的资产，必须使用尽可能比对于任何其它资产大的警惕性来加以保护。需要以更大警惕性保护数据这一论点证明起来非常容易，因为入侵者可能会采取人们无法想象的方式从互联网上的任何地方取得一个公司的数据。需要采取的措施包括：

- 对空闲的数据进行加密，这样，假如任何入侵者能够突破云提供商的安全防护措施，或者假如某个配置错误使得未经授权的人能够访问到该数据，数据就不会被破译。
- 对传送中的数据进行加密。设想数据将通过公用基础设施进行传递，并且可能会被其间的某些人看到。
- 要求在应用程序组件之间进行严格的身份验证，因而只把数据传送给已知接受方。
- 注意加密方法以及如何破解算法并随时间的推移用新算法取代。例如，鉴于 MD5 已证明容易受到攻击，那就使用一种像 SHA-256 这样的更加严密的加密技术。
- 对有权访问应用程序的人以及访问方式进行管理：
 - 考虑让管理员使用严密的、基于令牌的身份验证技术。
 - 对于客户登录/密码访问，考虑什么人管理身份验证服务器，以及访问是否在相应公司或云提供商的控制之下。

- 对于匿名访问存储器，例如，匿名 FTP，考虑客户是否必须通过云提供商进行注册后才能访问，或云提供商是否可以与公司的身份验证服务器联合起来。

网络安全做法

良好的安全做法贯穿于系统设计、实现和部署的各个方面。应用程序的设计必须是安全的，其接口只能给授权用户提供相应的数据。实现期间，开发人员必须认真避免可能导致易受如下技术攻击的编码做法：缓冲器过载 (Buffer Overflow) 或 SQL 注入 (SQL 注入)。部署时，应对操作系统进行硬化，并用最新安全补丁对软件各层进行及时更新。

在云计算中，应用程序部署在一个共享的网络环境中，而且使用非常简单的安全技术 (例如，VLAN 和端口过滤，对应用程序部署架构的各层进行分割和保护，并把客户彼此隔离开来。其中一些网络安全方法包括：

- 使用安全域把虚拟机组合在一起，然后通过云提供商的端口过滤功能控制对域的访问。例如，创建一个用于前端 Web 服务器的安全域，对外部世界只打开 HTTP 或 HTTPS 端口，并过滤从 Web 服务器安全域到包含后端数据库的域的流量 (图 11)。

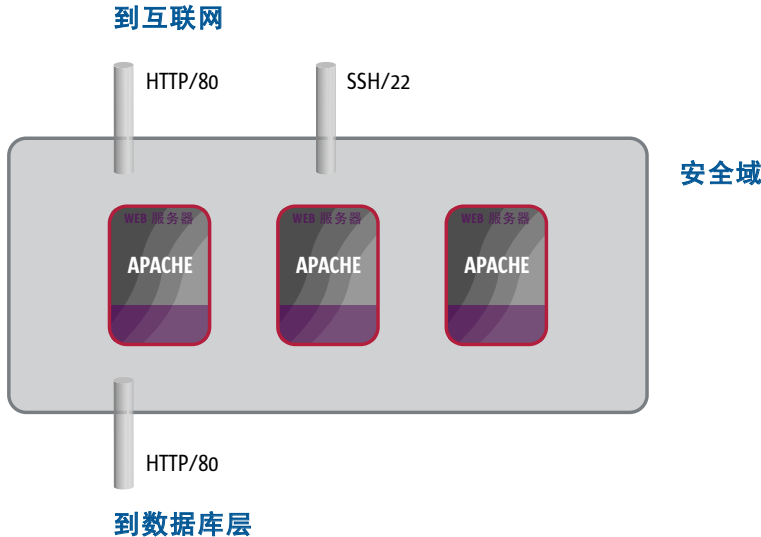


图 11. 云提供商应提供安全机制 (如安全域) 来保护一组虚拟机的安全，并控制流入和流出该组的通信流量。

有关 ISC 的更多信息，请访问：
<http://wikis.sun.com/display/ISC/Home/>

- 使用云提供商的基于后端的过滤来控制流量，或者通过在适当的地方插入内容交换机或防火墙设备来利用更加有状态的数据包过滤。为了对流量实现更加严密的控制，可以采用不可变服务容器 (ISC) 这一概念，它允许在单个虚拟机中部署多个软件层，并采用仅限虚拟机内部使用的预探测联网。此技术使用 Solaris™ Zones 在单个共享式 OS 平台上支持多个安全虚拟环境，并且只能使用 Solaris 操作系统和 OpenSolaris 操作系统。

第 4 章

Sun 公司与云计算

十几年来，Sun 公司一直致力于推动构成云计算基础的大型计算基础设施的技术进步。早在 20 世纪 90 年代，Sun 公司就已成为帮助服务提供商实现其大型网络的领先者，使他们可以向数以百万计的客户提供服务。金融机构和股票交易所利用大量 Sun 服务器处理各种任务：从处理交易到实施欺诈检测。Sun 公司通过开发一次一架 (Rack-at-a-Rack) 的部署模式、从光金属到应用程序的自动供应 (Automatic Provisioning)，以及具有每秒一 PB (Petabit) 吞吐量的大型虚拟网络，已经成为高性能计算行业的领先者。的确，通过按小时销售物理服务器时间，并帮助客户实现内部网络以支持其自己的业务活动，Sun 公司是云的前辈“网格计算”当之无愧的开拓者。

就像这些大型计算能力有助于 Sun 公司开发云计算解决方案一样，上述能力也有助于改善这些解决方案所需的系统品质：可扩展性、可用性、可靠性、易管理性和安全性。

来自 Sun 社区的创新

Sun 公司开发出用于云计算的基础性技术，并已成为这些技术所促成的社区开发流程的主要参与者。Sun 公司在长期保持 Solaris 操作系统的行业领先地位的同时，还围绕 OpenSolaris 操作系统发起一个相应的开放源运动。MySQL 数据库是首选的 Web 应用程序数据库，而 Java 编程语言驱动着全世界的网站和企业数据中心。基于社区的开放源 GlassFish 应用程序服务器提供一个 Java 软件执行容器，它已扩展为支持 Ruby 应用程序和 Drupal 内容管理系统。OpenSolaris Project Crossbow 已帮助扩展了 Sun xVM 管理程序中的多重租用支持。

在 Sun 公司大力促进的丰富而得到社区支持的软件下面，是使这些软件正常运行的功能强大的服务器、存储和网络产品，其中包括标准的可扩展 x86 架构服务器、Sun 公司 UltraSPARC® 处理器驱动的服务器产品线，以及包含 Sun 公司的高效率、芯片多线程 (CMT) UltraSPARC T1、T2 和 T2 Plus 处理器的服务器。Sun 公司 CMT 处理器处理高吞吐量工作负荷如此高效，以至于它们在各种内容负载平衡和应用程序提供产品 (如 Zeus 可扩展流量管理器) 中得到广泛应用。Sun 公司的开放式存储产品把开放源软件与行业标准的硬件结合起来，帮助减轻对高价位的专用型系统的依赖性。实际上，Sun 公司具有突破意义的 Sun Fire X4500 服务器帮助整个行业看到了把服务器和存储技术合并到同一系统中的优势。Sun 公司通过 InfiniBand 向采用 Sun Datacenter Switch 3456 (扩展到多达 13,834 个节点) 的大型计算网格，提供用于大型计算的虚拟网络技术。

要想了解该社区如何定义 Sun 云 API，
请访问: <http://kenai.com/projects/suncloudapis/>

社区与开放式标准

与一个社区合作创造出基于开放式标准的产品，并有助于提供投资保护。在一个像云计算这样的快速变化的新兴市场里，创建由于使用专有 API 和格式而被限制在一个供应商的云中的应用程序非常容易。使用开放式标准和开源软件可以最有效地保证：您现今创建的应用程序将来仍会有用，并将赋予您更换云提供商所需的灵活性。

Sun 公司所参与的开源社区开发他们所采用的开放标准，并且在开发出新产品时制定新的开放式标准。开放源、开放标准和开放 API 产生具有更大可移植性和使用寿命的应用程序。Sun 公司的开源社区凭证是无懈可击的，其项目包括：OpenSolaris OS、Linux OS、StarOffice™ 软件、NetBeans™ 平台应用程序框架、OpenSPARC™ 技术、Java 编程语言、Sun xVM 管理程序、Sun xVM VirtualBox、Sun 开放式存储解决方案、MySQL 数据库管理系统以及 Solaris ZFS™ 文件系统。

选择的重要性

Sun 公司的硬件和软件产品线与精品同义。Sun 公司提供对于基于 x86 架构的服务器的广泛选择余地，这些服务器由功能强大的 SPARC® 和 UltraSPARC 处理器所驱动，并且采用了 CoolThreads™ 技术。Sun 公司提供上述各种形式的选择，包括机架式和刀片式系统，为客户提供各种密度和 I/O 容量选择。Sun 公司为其各个服务器产品都提供虚拟化解决方案，包括在其 x86 架构服务器上支持 Sun xVM 管理程序、VMware vSphere 和 Microsoft Hyper-V，当然也包括您也可以选择操作系统，这包括 Solaris OS、Linux 和 Microsoft Windows。

选择云计算提供商

Sun 公司创新是云计算环境的基础，而云计算环境具有开放性，基于标准，且是社区成员共同努力的结果。加入 Sun 公司云计算社区就意味着您可以选择采用可发挥最大作用的服务器、存储和联网技术。这同时也意味着可以使用不为某个云提供商所拥有的软件栈、API 和标准，而是属于构建其云应用程序以拥有持久价值的公司。Sun 公司提供多种选择，这不仅仅是使用恰当硬件和软件组件完成工作任务方面的选择，而且包含利用云计算技术实现最大效益方面的选择。

要想了解 Sun 公司如何帮助构建您的云，
请访问: <http://www.sun.com/cloud/>

那些为了云计算加入 Sun 社区的机构可以拥有多种选项。Sun 公司可帮助各种机构构建自己专用的本地云，以便于将企业数据中心过渡到这种新的计算模式，同时保留对关键业务数据的绝对控制权。Sun 公司可帮助各种公司构建自己专用的非本地云，以便于利用成本低廉的新型大规模高效托管场所，例如，Switch Communications 公司的位于内华达州拉斯维加斯的 SuperNAP 托管设施。Sun 公司可以为那些希望成为云提供商的机构提供帮助，供应所需的硬件、软件和管理能力。而且，从现在开始，世界各地的机构都可以利用 Sun 公司公用云产品扩大其专用云—可以与 Sun 公司共同位于 SuperNAP 站点，并享有高速度本地基础设施所带来的优势，也可以使用 Sun 公司在互联网上提供的

服务。无论是您在寻找用于开发和测试的云计算，寻求体验在云中托管应用程序，卸载特定功能，还是将云用于超负荷计算，Sun 公司都可以得心应手地帮助企业构建并利用云计算。

感谢

本白皮书的完成主要归功于 Jason Carolan 和 Steve Gaede 的努力。其他做出贡献的人士还包括 James Baty、Glenn Brunette、Art Licht、Jim Remmell、Lew Tucker 和 Joel Weise。此外，还要感谢 Benoit Chaffanjon、David Douglas、Mikael Lofstrand、Ken Pepple、Scott Mattoon 以及 John Stanford 所提出的宝贵意见。

本页故意留空。

本页故意留空。



Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN (9786) Web sun.com

© 2009 Sun Microsystems, Inc. 保留所有权利。Sun、Sun Microsystems、Sun 徽标、CoolThreads、GlassFish、Java、MySQL、NetBeans、OpenSolaris、Solaris、StarOffice、SunTone、ZFS 和 The Network Is The Computer 都是 Sun 公司或其在美国和其他国家或地区的商标或注册商标。所有 SPARC 商标都为授权使用，并且都属于 SPARC International 公司在美国和其他国家或地区的商标或注册商标。带有 SPARC 商标的产品基于 Sun 公司开发的一个架构。UNIX 是一个在美国和其他国家或地区使用的注册商标，并通过 X/Open 有限公司独家授权使用。此处包含的信息可能随时更改，恕不另行通知。
SunWIN #504162 文献#GNWP14947-0 06/09