



云计算标准及性能评估

云计算标准及性能评估

从应用运行的角度来讲，云计算的性能就是网络性能、应用性能以及云计算基础架构性能的总和。云供应商只能对最后一个因素负责，而不能对前两者负责。因此，用户在衡量供应商云服务质量时必须遵循一个合适的标准。本文介绍制定云计算服务标准必要性，为了满足这一标准而需要满足的配置，以及云计算服务等级（SLA）的制定和遵从问题。

云计算标准

在部署和扩展云计算方面遵循标准非常重要，软件服务会从所有标准化开发中受益，因为它们能够确保应用程序间的互操作性良好，不过遗憾的是想要扩展功能，同样会受到供应商的锁定和限制。

- ❖ 云计算标准使部署和扩展服务更自由

云计算性能

对于希望削减硬件投资成本的公司来说，云计算是首当其冲的选择。其收付自助型模式可应用于诸如 Netflix 和 Target.com 等在线业务，相关开发人员正在致力于这方面的测试和开发。

- ❖ 云计算所面临的性能问题
- ❖ 如何评估云计算性能？

云计算 SLA

在网络服务中服务等级协议（SLA）是很常见的，它们常常被用来表述网络服务的限制因素——通常被称为“QoS”（服务质量）。但是，在云计算或平台服务中，我们就很难找到可用的指标来作为 SLA 谈判的标准。

- ❖ 关注云计算中的服务等级协议（SLA）
- ❖ 满足云计算的性能标准和 SLA

云计算标准使部署和扩展服务更自由

目前，已经有一系列机构在致力于研究软件即服务（SaaS）中服务之间的互操作，如 Internet 标准化团体（W3C）负责管理 XML 和 WSDL 等标准，OASIS 负责定义 WS-* 标准过程，这些积极的举措有助于减少服务提供商及其客户的风险。只要遵循这些标准，应用程序就不会被限死在一个特定的技术下，标准化促进了软件即服务模式的发展。如果每个服务提供商都制定自己的标准，那用户也就会受限于这个提供商采用的技术。反之，如果每个服务提供商都遵循统一的标准，那服务之间的互通就没有问题，购买或托管服务的用户也不会受限于一个单一的提供商。

一旦软件跨过服务这道屏障成为服务，应用程序间互操作问题就应备受重视。另外如果最初设计的硬件环境不能满足作为服务后的应用程序运行时需要，用户购买的服务将会受到严重影响，在购买软件即服务的应用时，我们必须重新分析数据中心或托管提供商的基础设施。

即使对于一个非服务型设计，部署和扩展应用程序时如果超出其初始阶段的设计余量，这时需要同时考虑硬件和软件环境扩容，需要考虑是否虚拟化操作系统，是否采用集群中间件产品实现负载均衡，以及对应用程序进行定制修改，以适应不断增长的需求。

在软件即服务模式下，除非是实力雄厚的组织，要建设这种基础设施可能会让人望而却步。但这样的服务提供商已经出现，它们允许小型组织按需扩大应用能力，即使用多少就支付多少费用，即按使用付费模式，这些服务提供商包括亚马逊 EC2 服务和谷歌的 App Engine 服务，以及一些专业化的软件公司，如 3Tera，RightScale 和 Elastica 等。

在部署和扩展云计算方面遵循标准非常重要，软件服务会从所有标准化开发中受益，因为它们能够确保应用程序间的互操作性良好，不过遗憾的是想要扩展功能，同样会受到提供商的锁定和限制。

在当前这个锁定是不可避免的，因为按使用付费模式需要同时购买硬件和软件的组合，并且有不同的级别，服务提供商不可能让你无限制地扩展。目前统一费用模式仍然是最流行的，在这种模型下，托管供应商通常提供相同的操作系统和硬件环境，虽然用户托

管软件服务时有多种选择，但仍然不方便，一方面担心购买的托管环境不能满足需要，另一方面担心钱花多了。

在解决标准化问题方面第一个有意义的举措是 2007 年由分布式管理任务组（Distributed Management Task Force）创建的开放式虚拟机格式规范（Open Virtual Machine Format Specification，OVF），这个规范得到了包括戴尔，惠普，IBM，微软，VMware 和 XenSource 在内的支持，这个标准允许开发人员安装预配置的应用程序，可以很容易地复制它们，扩展起来就很方便，也没有专用硬件和软件架构的约束。

虚拟机技术使多个操作系统可以运行在同一台服务器上，许多按使用付费的提供商在它们的架构中采用了类似的做法。与 OVF 有关的创新性项目 Kensho，它是一个开源工具，基于 OVF 标准，它可以导出/导入实例到各种虚拟设备中，只要这个虚拟设备遵循 OVF 标准即可。

关于云计算一个更全面的标准是由 3Tera 承担的 Cloudware，它和 OVF 不同，它集中在单个虚拟机上，Cloudware 的重点是精简云环境，如精简数据库集成和复制。

虽然 Cloudware 还处于起步阶段，它已经引起了许多按使用付费提供商的注意，这个工具有可能使它们摆脱目前使用的零碎的方法，例如，亚马逊 EC2 服务使用了亚马逊机器镜像（Amazon Machine Image，AMI）的概念，谷歌 App Engine 使用了强制 CPU 和数据配额。虽然 Cloudware 是由 3Tera 开发的，RightScale 和 Elastica 也提出了各自不同的解决方案，它们允许用户部署和扩展应用。

如果应用程序是按使用付费模式设计的，目前你能改变的余地很小，因为每个提供商的硬件和软件架构都不一样。

因此，如果你的应用程序能从按使用付费云计算提供商的部署和扩展功能受益，你应该仔细研究一下它们遵循的是什么标准，否则你的应用程序和你的组织就会依赖于一个单一的供应商。

[查看原文](#) (作者: Daniel Rubio 译者: 黄永兵 来源: TechTarget 中国)

云计算所面临的性能问题

对于希望削减硬件投资成本的公司来说，云计算是首当其冲的选择。其收付自助型模式可应用于诸如 Netflix 和 Target.com 等在线业务，相关开发人员正在致力于这方面的测试和开发。

但是现在很多人对此持保留意见，许多人对于云计算的承诺和夸大性宣传都持认真理性的态度，还有一些人想知道云计算究竟是不是他们所需要的东西。来自澳大利亚新南威尔士大学的一项研究项目表明，当前云中存在诸多弱点。2009 年 1 月以来，Anna Liu 博士对 Google 应用引擎、微软 Azure 和 Amazon 网络服务（AWS）进行了基本的测试。

“对于这些大型分布式系统，随之而来的是许多不确定性，”大学系统架构和规划专家 Liu 博士说。Liu 说她创建了一个简单的网络应用，用于测试每个云服务的响应时间和其他功能。她表示 Azure 的结果是最不可靠的，这是由于该服务仍在测试阶段。

“我们注意到随着时间的推移，云服务的性能有着相当显著的变化，”她补充说，“很明显不同云是为不同用途而建立的。例如，Google 应用引擎只是为了简单、快速网络应用而构建的；任何花费时间超过 30 秒的计算请求将会导致高达 0.07% 的错误率。”与此同时，AWS 在针对其数据库产品 SimpleDB 问题的测试中在响应时间和适方面也存在高度差异。

Amazon 正试图更多地了解 Liu 的研究，因为该研究还未公开。“我相信这个特定的测试应用在极短的时间内向 Amazon SimpleDB 发出了大量的服务请求，试图让系统过载。”Amazon 发言人 Kay Kinton 在一封邮件中如此说。她表示 SimpleDB 返回了一个“服务不可用”的消息，以此作为一种保护措施。Kinton 也说她相信 SimpleDB 能够扩展处理“现实生活和工作中的应用”。

Liu 认为她的测试应用并不是不符合使用 SimpleDB 的实际应用。她还表示，其错误率与 Amazon 数据中心在七月的停电也有关系。

同样，如同她在测试 Google 应用引擎时遇到通信繁忙一样，Liu 说她还撞上过 Google 阻塞服务的保护。她将其与因二月 Victoria 森林大火而推出的网站相比较，该网站同样基于 Google 应用引擎。Google 工程师不得不在用户因试图了解火情而大量访问网站导致其崩溃后对网站进行修复。

“我们的研究只是希望了解下一步要做什么，从表面上来了解云的主观限制和性能缺陷。”刘说。用户可能不知道他们将于何时何地遭遇阻力或障碍，这使得他们很难对问题进行规划。Liu 表示，她的研究并不是结论性的，它只是对云计算现状的一个及时的反馈，并将适应不断增长的需求。她说她的研究并不是企图“为云制定标准”，因为供应商对这一点很明确，他们的研发会针对不同的应用。

“如果将 Amazon 服务器与 Google 的编程平台进行比较，全球基准将是毫无意义的，”她说。但是 Liu 认为要制定精确描述云中性能变化的独立性能指标是毫无意义的。“不久的将来一定会有标准的商业应用产生。”她说。

“这些研究揭露了一些实际问题以及云计算的复杂性问题，”澳大利亚公司对象咨询的系统架构师 Kevin Francis 说。他认为这种研究对于揭露云模型的限制因素是必要的。他补充说，Liu 的研究只是揭示了云计算使用规划方法的冰山一角。“它表明云是一个美妙的东西，您可以将任何东西放置云中，这没有问题。”他说。

云用户在充满缺陷的环境中工作

很显然，云计算用户已找到了一些处理问题的方法。Forrester 基础设施分析师 James Staten 说，Amazon 和其它公共云资源的开发显然更多地是在考虑成本而不是性能，但是开发人员通过使用云的显著特点已默认了其不确定条件——即时自助服务的特点。

“云开发人员经常会这样做，将一组信息连续发送四次，找出最快的那个。”他说。开发人员知道这具有不确定性，因此他们将建立一个虚拟实例数组，找出性能最佳的那一组，然后将其余的关闭，这是一个很老土的做法，但是所需成本很少。这对成长中的私有云市场产生了影响，他说。

Staten 表示，那种不确定性对于 IT 企业并没有吸引力。他说类似 Liu 博士的这种性能研究只是降低了用户接受公共云资源的信心，而他们对安全和管理问题的日益关注则更加有利于私有云的发展。

[查看原文](#)

(作者: Carl Brooks 译者: 滕晓龙 来源: TechTarget 中国)

如何评估云计算性能？

云计算性能怎么样？这要看你问的是谁。有三种云计算模式；你使用哪一种模式，决定了会得到什么样的性能。使用软件即服务（SaaS）系统、处理标准 Web 时延的那些人无法告诉你性能方面的太多信息。不过，使用高级“大型数据”系统的那些人会告诉你大不一样的情况。

你需要考虑几种性能模式；性能模式可以分为三个非常基本的类别：

- 面向客户（性能方面有不足）
- 面向云（性能方面有优势）
- 混合型（性能不好说，要看实施情况）

面向客户的云计算架构是指这些系统：云计算提供商（通常是 SaaS 提供商）通过互联网，与用户不断进行交互。这里的问题倒不是云计算提供商提供服务的速度缓慢，而是 SaaS 提供商与浏览器之间不断出现的机器对机器后端会话存在时延。

除了在贵公司与 SaaS 提供商之间自行建立一条成本高昂的专有链路，你对此基本上无能为力，但是这么做马上会降低 SaaS 具有的价值。面向客户的平台其速度显然不如本地网络上运行的应用程序一样快；但在大多数情况下，除非出现网络饱和，否则用户是不会注意到这种时延的。

面向云的云计算架构是指这些系统：处理过程在云内部进行。大多数基础架构即服务（IaaS）提供商和一些平台即服务（PaaS）提供商就属于这一类。这种系统通常能够提供优于内部部署系统的性能，原因在于它们可以访问多得多的虚拟化资源，而且能够动态分配这些资源。

比方说，你可能有一个数据库查询导致内部系统出现饱和，需要多个小时才能运行完毕。相比之下，需要增添多少台服务器来支持急剧增加的处理量，配置充足资源的云计算提供商就能提供多少台服务器来满足要求，可以在几分钟之内返回结果集（result set）。这体现了云计算具有的灵活扩展性，也是权衡考虑性能与成本时，云计算具有的一个显著卖点。

大家可能也猜到了，混合型云计算模式充分利用了面向客户的云计算模式和面向云的云计算模式，通常用于平台即服务系统，但是这两种云计算模式所占的比重有所不同。其概念是，你可以将用户交互与高度扩展、随需应变的后端处理进行混合搭配。缺点是，你与浏览器的联系越频繁，给这种模式带来的时延就越长。因而，面对这种构架的解决办法就是，对基于云的后端服务器与浏览器之间的交互进行优化。你可以利用一些老式的分布式应用架构来做到这一点。

如果你正在向云计算迁移，别忘了考虑其性能有好、坏、中之分。

[查看原文](#)

(作者: *Kevin Beaver* 译者: 黄永兵 来源: TechTarget 中国)

关注云计算中的服务等级协议（SLA）

许多 IT 经理正在考虑把许多应用及服务迁移进云端。一部分人因为经济原因被迫考虑云计算，而另外一部分人考虑提供一些新的 IT 服务。不管怎样，IT 经理目前以及不久的将来不得不面对服务等级协议（SLA）。

对于许多 IT 经理来说，评估 SLA 是不容易的。毕竟大多数的 SLA 都是一些条款形式的内容，人们很难确定某个运营商实际能够提供哪些服务。而且 SLA 的提出主要是为了保护运营商的利益，而不是针对客户，这样使整个事情变得更为复杂。许多运营商提供 SLA 主要是为了避免一些不必要的纠纷和诉讼，同时提供给客户最小限度的保证。也就是说，当其企业选择了一个云运营商并且对那些服务进行有效的安排之后，SLA 同样能够成为 IT 经理一种有效的工具。

IT 经理需要关注 SLA 的三个方面：数据保护，连续性和费用开销。毋庸置疑，数据保护是最需要关注的一个要素。IT 经理想要确认谁有权使用这些数据。刚开始确定数据保护的级别似乎很容易，但是还是有很多隐藏的问题。IT 经理必须查出这些问题并且解决他们。

这些问题进一步的升级就涉及到了知识产权保护问题。所有的问题最后都归结为谁最终能够控制客户的这些私有数据。

一个 IT 经理需要明白如何利用运营商的基础构架和服务来为那些必须的应用和数据提供连续不断的保护。业务不间断性非常重要。最理想的情形是运营商保证提供 100% 的不间断服务，但实际上这样的保证是不可能实现的。

所有的服务提供商都将会经历在某一时刻宕机的情况，因为有很多超出他们控制范围的情况发生，包括自然灾害以及社会生活中发生的一些不确定因素。大部分的服务提供商最多能够给予 99.5% 正常运行时间的保证。但是这些保证通常还附带另外一些限制条件。即便如此，运营商可以尽力保证这些服务在一个可接受的层次范围之内。接着我们能得出这样一个问题，当服务中断的时候究竟怎么办？

综上所述，这些问题的答案就是运营商将能够提供的服务等级。更重要的就是一个 IT 部门怎样处理这些故障，这些故障又将会影响到哪些用户。

一些运营商都将价格要素包含在他们的 **SLA** 中，其余的则将这些费用放置在一些独立的合同条款中。不管怎样，**IT** 经理必须明白这些费用包含在基于云的服务中。这些费用不仅仅和预算有关，而且通常被用来确定投资收益率。价格分析通常都是财务部门的任务，但是 **IT** 经理能够帮助加速这个过程，或许还能够为那些花费在云服务上的费用提供一些简单合理的解释。

找到这些问题的答案并牢记以上要点能够帮助 **IT** 经理作出一些有理有据的决定。当他们选择一个服务提供商并且打算和提供商建立长期合作的时候，这些决定能够同时保证服务的有效以及可靠。所有这些归结起来都是为了简化关于服务等级协议（**SLA**）的描述，并且给大家提供 **SLA** 的一个通俗概念。

[查看原文](#)

(作者: Frank Ohlhorst 译者: 胡彬 来源: TechTarget 中国)

满足云计算的性能标准和 SLA

在网络服务中服务等级协议（SLA）是很常见的，它们常常被用来表述网络服务的限制因素——通常被称为“QoS”（服务质量）。但是，在云计算或平台服务中，我们就很难找到可用的指标来作为 SLA 谈判的标准。

从整体上来看，这两者遇到的问题是一样的——你必须制定一个服务质量标准，如果没有达到标准就要想办法去改善它。最大的差别是在细节方面，你肯定要从用户的应用体验限制因素下手去解决这一问题。对于企业而言，利用云计算是出于提高可用性范围和系统性能的目的，这也正是 SLA 必须要解决的问题。

在云计算 SLA 中最重要的一点是要明白所有与应用体验相关的东西都不属于云计算。从应用运行的角度来讲，云计算的性能就是网络性能、应用性能以及云计算基础架构性能的总和。云供应商只能对最后一个因素负责，而不能对前两者负责。因此，在编订 SLA 时要明白其它的两个因素究竟会对云计算的整体性能起到什么作用，这一点很重要。

通过 Internet 或其它服务来访问云计算服务将很难去创建一个合适的云计算 SLA，因为网络中的延迟、丢包或传输失败率完全是动态变化的。如果你想让用户所体验的传输/应用性能得到保证，你就需要采取一些办法来限制这些变化。如果你的云供应商可以拥有专用的 ISP（网络服务供应商）与其直连，你就可以和他进行 SLA 谈判。如果你想要从多个地点或 ISPs 访问云应用，那就很难制定一个合适的应用性能标准。

如果你将应用性能的变化排除在外，你就可以利用本地的服务器和网络资源来运行应用，在理想的环境来对应用进行衡量。这其中应该包括内存、存储的变化会对性能产生哪些影响，因为这些因素在云计算服务中也是经常变化的。此外，你最好尽可能多地对云中的 IT 资源进行复制来测试性能，这点很重要。

当处理完应用和网络性能因素之后，就可以用所得到的信息来设定云计算性能标准了。例如，如果一个应用在本地运行会有 1 秒的传输响应时间，还有 0.5 秒的网络双向连接延迟（对 Internet 或 VPNs 来说有些不合理），总共的性能延迟时间就是 1.5 秒。如果你的业务部门想要将系统响应时间保证在 2 秒之内，那留给云计算的延迟时间就只有 0.5 秒。

下一个重要的步骤是将应用性能转换成一套可以在云供应商的基础架构内进行测试的标准。只有你为云服务进行了专用的配置，这才能发挥作用。因此，你需要和你的云供应商（或候选供应商）通力合作，来设计一套最佳的云计算配置满足自己的需求。

其中包括，你是否需要使用预留的、或是专用的云资源，你所运行的应用每次需要启动的镜像文件数量，他们所需要的运行环境，所使用的数据库、系统类型和内存大小等等。你还需要对这些配置进行测试，确保它能满足应用用户的基本性能需求。此外，这种配置测试有助于你确定与系统性能和可靠性直接相关的云的特点——比如说从失败的应用案例或案例的负载均衡角度。

当前，你必须利用云供应商的管理工具或能力在这套配置当中创建一系列的资源使用、应用可靠性及性能指标。理想的状态是——如果这些标准都能得到满足，配置就可以按照你的设计运行，应用用户的业务目标也可以实现。

记住，尽管说你制定云计算 **SLA** 的目标是保证传输或应用的性能，但这也是你自己的责任，你需要创建一套用户体验的性能标准，然后利用它来制定云计算和网络的性能目标。

为了对你的云计算 **SLA** 实现高效的利用，你需要配置故障隔离工具，将故障与网络或云中的应用隔离开来。同时，你还需要将其与你从云计算供应商那里得到的云管理工具整合起来创建一套监控机制，利用它对系统性能进行主动监测并对用户的控诉给予反馈。

[查看原文](#)

(作者: Tom Nolle 译者: 涂凡才 来源: TechTarget 中国)