

基于 Windows Azure 的大规模 Web 应用程序设计

卢朝晖

(海南师范大学 网络中心,海南 海口 571158)

摘要: Windows Azure 作为微软公司赢在未来的云计算平台,具有许多新的特性,正逐渐受到业界的青睐,因此在现阶段研究基于该平台来设计和开发大规模 Web 应用程序十分必要。首先分析 Windows Azure 云服务中与编程相关的关键组件,如计算服务、存储服务和数据库服务的特点,特别是它们与本地开发的差异;在此基础上根据平台提出两个设计大规模 Web 应用程序的要点:计算资源的“无状态”处理和数据库的“横向”扩展,并给出了相应的实例进行说明。

关键词: Azure;大规模;Web 应用程序;设计

中图分类号: TP39

文献标识码: A

文章编号: 1673-629X(2014)12-0057-04

doi: 10.3969/j.issn.1673-629X.2014.12.014

Design of Large-scale Web Application Program Based on Windows Azure

LU Zhao-hui

(Network Center, Hainan Normal University, Haikou 571158, China)

Abstract: Windows Azure as Microsoft leading the future market of products has many new features and is being paid more and more attention by the industry, so in the present the study of designing and developing large-scale Web application is very necessary. Firstly, analyze the characters of the key programming components of Window Azure's cloud services, such as computing service, storage service and database service, especially understanding their differences in the development environments between Azure and local server. Then, according to the platform, put forward two important design points of developing a large-scale Web application, that are making computing resources stateless and scaling out database. And some corresponding examples are given.

Key words: Azure; large-scale; Web application program; design

0 引言

Windows Azure 作为微软公司赢在未来的公有云计算平台^[1-2],因其采用业界顶尖的云计算技术,熟悉 ASP.net 应用程序或 WCF 服务的开发者不需要大幅度地更新知识和经验,以及应用程序可以在云、数据中心和本地服务器之间方便地迁移^[3]等优势,且随着微软数据中心部署规模在全球的不断增加,正逐渐受到企业和用户的青睐^[4]。例如,宝马公司利用 Windows Azure 结合 Facebook 社交网络和客户关系系统开展市场营销;PPTV 亚洲电视网将 PPCloud 的核心技术迁移至 Windows Azure,即客户上传的视频内容通过 Blob 存储进行保存,并由 Windows Azure 在全球的 CDN 网络将内容送达离观众最近的节点,确保视频播放速度;金蝶软件利用 Windows Azure 部署 K/3

Cloud,为大中型企业 ERP 系统快速上线提供了可行性解决方案,同时为它们节约了大量硬件和运营维护成本;文献[5-8]基于 Windows Azure 开发了实验学习系统、煤炭物流车辆调度系统等。因此,在现阶段了解 Windows Azure 在基础平台和服务行为方面的特点,研究如何根据平台进行设计,对于编写能够在云中运行的大规模 Web 应用程序具有十分重要的意义。

1 Windows Azure 的特点

Windows Azure 可提供三种不同层次的使用模式^[9]:虚拟机(Virtual Machines)、网站(Website)和云服务(Cloud Services)。其中云服务(以下 Azure 均特指云服务)属于“平台即服务(PaaS)”,是一个以 Windows Server 2008 为核心打造的,专属于数据中心使用

收稿日期:2014-03-26

修回日期:2014-06-30

网络出版时间:2014-10-23

基金项目:海南省自然科学基金(612126,612122);海南省高等学校科学研究项目(Hjkj2011-26)

作者简介:卢朝晖(1974-),男,海南海口人,副教授,硕士,研究方向为计算机网络及应用。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20141027.1429.006.html>

的应用程序运行环境。对开发者而言,它隐藏了云中的硬件、网络架构、操作系统、编译系统等,提供了虚拟化的计算资源、可扩展的存储服务、丰富的开发工具包(SDK)、易于部署和运维的 Web 管理门户网站(Portal)等^[10]。云服务中与开发者编程相关的组件有:

1.1 计算服务

云服务提供了 Web 角色(Web Role)和辅助角色(Worker Role)两种计算单元^[11],其中 Web 角色可以运行.NET 或开源的 PHP、JSP、Python 等代码,由 Azure 中的 IIS 负责执行;另一种是辅助角色,可以是托管或原生(native)代码,由 Azure 中的代理进程(Agent Process)负责执行。这两种角色具有如下特点:

(1) Azure 目前提供特别小、小、中、大、特别大共五种不同规模的计算单元,但无论哪种规模都有一定量的 CPU、内存、存储空间和带宽资源。

(2) 每个建立在角色之上的实例(Instance)都是非持久的。主要表现在:当某个实例由于某种原因出现故障,Azure 会自动检测这一行为并立即产生一个新的、具有初始状态的实例来代替它,也就是说之前发生在实例中的任何变化不会在新实例中体现;当应用需要扩容以满足需求时,Azure 可以在现有的部署中动态增加新的实例,此时新实例仅具有初始状态,如果其他现有的实例在运行过程中发生改变,那么它们的状态可能不一致;当应用需要收缩时,不是由用户决定在现有的部署规模中关闭某个或者某些具体的实例,而是由 Azure 在后台自动执行。

1.2 存储服务

云服务提供了三种非关系型的存储服务,分别是 Blob、Table 和 Queue。

(1) Blob。

Blob 在 Azure 中提供可用性高的方法来存储非结构化数据,如二进制文件、文本文件等。其中,区块型 Blob 以大小不同的区块(最大 4 MB)来存储数据,支持流式访问,并行上载,同时在写入时需两阶段确认,以决定是否客户端进行重传操作,因此适合通过网络高效传输大型文件;分页型 Blob 以连续的地址方式来存储数据,在访问时可随机读写,而且写入操作会立即被认可。

(2) Table。

Table 在 Azure 中为列数据(二维的结构化数据)提供持久、可扩展性高的存储。Table 提供了 {partition key, row key} → {data[]} 语义来存储和访问数据。换句话说,一个表是一个实体的集合,保存了一组实体的信息,每一个实体记录是一个行记录;一个实体是该实体的关键属性和相关属性的集合,每一个属性是一个名/值对,对应一列。

(3) Queue。

Queue 一般用于在多个并发表务器和订阅服务器之间提供中转消息传递服务。Queue 在语义上不像传统消息队列,如具有先进先出,每次只传递一条消息等功能。它的工作流程是:消息入列;消息出列;消息被处理,这时被处理的消息对其他访问者在一段时间内(缺省是 30 秒)来说是屏蔽的;如果消息被成功处理,那么将被访问者显式地删除;否则等屏蔽时间过后,该条消息将重新显示以供其他访问者访问。这也就是说,队列中的消息可能出现多次,从而保证消息至少被处理一次,当然这就不能保证消息出现的顺序了。

以上 3 种云存储服务具有以下特点:

(1) 每个存储帐户是一个可扩展且具有一定容量的基本单元,超过单个帐户容量限制就要求在同一个应用程序中利用多个帐户。

(2) 提供持久数据服务,且均可通过基于 HTTP 的 REST API 供外部程序访问^[12]。这意味着在任何平台下的任何程序语言,只要能够支持 HTTP 通讯协议,即可借此一开发接口存取云存储服务,具有很强的解耦特性。

(3) 所有写入或更新云存储的操作都在三个存储节点间透明、一致地复制(这三个节点处于不同的升级域和故障域中),从而提供可用性。

1.3 SQL Azure

Azure 中还提供了关系型数据库服务—SQL Azure,支持应用程序快速设置关系数据库并向其增加、删除、更新以及查询数据。SQL Azure 提供许多熟悉的但并不与 SQL Server 一一对应的功能和特性,同时精简了软硬件配置、故障恢复等任务,其目的在于满足专门适用于云应用程序的一套不同的要求(通过弹性扩展实现无限空间、高可用性、降低维护成本等)。值得注意的是,SQL Azure 是一种运行在共享基础结构上的多租户服务,除了来自不同租户的数据库共享基于商用硬件构建的底层物理节点,其他系统用户也可以使用同一底层基础结构上的关键资源(如工作线程、事务日志、I/O 等),因此单个数据库所能支配的资源限定在一定范围之内,而一旦超出这些限制,在租户或物理节点级别,SQL Azure 就会中止使用或断开连接。

2 大规模 Web 应用程序的设计要点

构建较大型应用程序以满足大规模用户访问的传统方法是依赖于将向上扩展(购买较大的多核心/大内存系统、数据库服务器以及构建较大型的数据中心等)与向外扩展(增加无状态的 Web 或应用程序服务器节点)相结合^[13]。但在 Azure 中,因为不论计算资源还是存储资源的规模都是有限的,因此随着事务处

理量的增长,实现真正可扩展应用程序的唯一途径只能是向外扩展。这就需要从以下两个方面设计基于 Azure 的 Web 应用程序。

2.1 计算资源的“无状态”设计

实现计算资源无状态的关键是避免将持久数据存放在实例的本地存储(local storage)中,而将它们存放在实例之外,如云存储服务、SQL Azure 等。如上所述,实例都是非持久的。如果将持久数据保存在实例内,当实例因某种硬件原因而重新恢复,或因需求增减而动态增加或减少实例数时,应用中的实例会导致不同步或者不一致。以 Session 的处理为例,许多 Web 应用需要使用 Session 在无状态的 Web 连接中跟踪会话的状态。在本地部署中,Session 一般保存在服务器的内存里,但在 Azure 中,由于 Azure 的负载均衡是无状态的,因此不能将其保存在实例内,可保存到具有较高可扩展性和可靠性(备份在 3 个地方)的 Table 中,如果要考虑读写效率,也可将其保存在 Azure Cache 服务中。Cache 服务可跨多个 Cache 服务器在内存中缓存数据,因此具有较高的性能,但是它仅支持 .NET Framework 4.0。

其次,通过松散耦合的通信来实现计算资源之间的组合。图 1 是一个典型的基于云服务的 Web 应用程序编程模型。

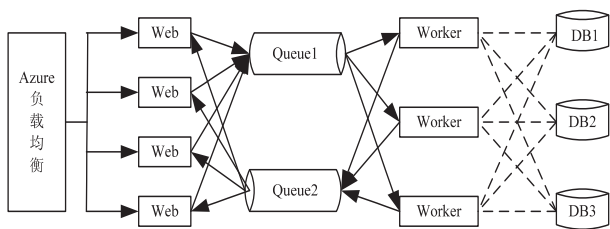


图 1 典型的 Web 应用程序编程模型

应用程序的前端和后端分别由多个相同的、无状态的 Web 角色实例和辅助角色实例组成,前后端之间通过 Queue 进行通信。其中 Web 角色群通过 Azure 负载均衡器循环接收用户发出的 Http 请求;Web 角色将用户要执行的任务放入 Queue1;辅助角色群通过轮询读取队列 1 中的信息在后台执行用户的任务,并将执行结果放入 Queue2;Web 角色从 Queue2 读取结果并返回给用户。这一模型的优点在于:

(1) 通过设置配置文件就可以动态地对角色或辅助角色实例数进行缩放,以适应不同规模的用户访问,同时当某个实例出现故障时,也不影响用户的访问。

(2) Web 角色实例与 Queue、辅助角色实例与 Queue 之间通过 Rest API 进行通信。这种通过利用云存储服务 Queue 来解耦前后端计算资源的松散耦合通信,确保了弹性扩展的可能,即任何一方的扩展不影响到另一方。

2.2 数据库的“横向”设计

根据 CAP 理论—Web 服务无法确保所有下列的三个属性同时满足:一致性(客户感知到一系列操作能够一下子执行完毕)、可用性(每一个操作必须有意想中的响应并终止)和分区耐受性(操作均能够执行完毕,即使个别组件不可用)。显然随着 Web 应用规模的扩张,传统数据库的 ACID 特性—原子性(Atomicity)、一致性(Consistency)、隔离性(Isolation)、持久性(Durability),是不合适的。因此为了提高可用性,BASE—(Basic Availability:基本可用;Soft-state:软状态/柔性事务;Hard state:面向连接;Eventual consistency:最终一致性)是一种很好的选择^[14]。在具体实施时主要考虑以下问题:

首先对数据进行扩展,提高数据库的可用性。如前所述 SQL Azure 可以轻松交付大量的相对较小的数据库,因此利用它在 Azure 上实施高度可扩展的应用程序要求采用横向扩展方式,从而可以结合使用多个数据库的资源来满足存储容量扩张的需求。横向数据扩展有两种策略。第一种策略是功能扩展,即根据功能进行数据分组以及对这些功能组进行跨数据库部署。例如,对于为不同客户群服务的多租户应用程序,该应用程序可以为每个客户都创建一个数据库;或者在一个购物网站中,可用一组数据库主机存放用户数据、一组存放商品数据、一组存放购买数据,等等。即使将功能一一解耦,单项功能的资源需求随着时间增长,仍然有可能超出单一数据库的能力。这时就要采用分片扩展策略,即将属同一功能域内的数据进行跨数据库的拆分。分片扩展可沿二个维度进行:一个是水平分片,例如,用户表可按分区值(如用户的 ID 或经过哈希函数处理的邮箱等),通过分区函数映射到若干个数据库中;另一个是垂直分片,例如,用户表中核心的用户信息(姓名、身份证、联系电话)存储在某个数据库中,相关信息(如虚拟形象图片的 URL)则存储在另一个数据库中。

其次,在提高可用性的同时,要实现最终一致性。在电子商务网站中,某件商品也许在卖的过程中,每个消费者在下单时看到的库存数是不一样的,但当商品最终卖完时,若有消费者在此时付款并提交订单时应提示“库存数都为 0”,这就是所谓的“最终一致性”。在单一数据库中,可依托数据库本身提供的约束机制(如外键、原子操作)来在不同表之间保持一致性,这对数据库部署策略来说是一个耦合的模式,随着事务量的增长这将不利于进行水平扩展。这就需要把数据约束从依赖数据库转到依赖应用程序本身。也就是说,要了解应用程序及其确切的业务要求,然后慎重设计和实施客户端代码来解决“最终一致性”。下面给

出一个例子来简要介绍如何在数据库表之间提高可行性同时实现最终一致性。

示例模式有两个功能组:用户表 `user(id, name, sell_sum, buy_sum)` 和交易单表 `transaction(xid, seller_id, buyer_id, amount)`。每次卖出一个物品,将会往 `transaction` 表添加一条记录和更新买方和卖方的计数。若 SQL 使用 ACID 式的事务,则对应的形式化 SQL 语句如下:

```
Begin transaction
insert into transaction ( xid, seller_id, buyer_id, a-
mount); //写入一条交易记录
update user set sell_sum=sell_sum+amount where id
=seller_id; //更新卖方的销售总量
update user set buy_sum=buy_sum+amount where id
=buyer_id; //更新买方的购买总量
End transaction
```

上述三个数据库操作通过外键进行关联,同时被看作是一个原子操作,任何一个操作不成功,整个操作就会被撤销。下面通过 `Queue` 来进行数据库之间的解耦,提高可用性,同时又利用 `Queue` 的这一持久消息队列机制的特性来保持最终一致性。形式化的程序代码如下:

```
Begin transaction
Insert into transaction ( xid, seller_id, buyer_id, a-
mount); //写入一条交易记录
Queue message " seller, seller_id, amount"; //更新
卖方销售总量的指令入列
Queue message " buyer, buyer_id, amount"; //更新
买方购买总量的指令入列
end transaction //这时即使 user 表出现问题,也不
影响整个操作
for each message in queue
Begin transaction
Dequeue message //更新指令出列
if message[0] == seller
update user set sell_sum = sell_sum + message[2]
where id=message[1]; //更新卖方的总量
else
update user set sell_sum = sell_sum + message[2]
where id=message[2]; //更新买方的总量
endif
end transaction
end for
```

这个例子仅用来简单演示,如果通过程序设计而不是依赖数据库的约束条件来保持一致性,它并没有解决所有的问题。例如,因为 `Queue` 不具有先进先出

的特性,系统不能保证以接收的先后顺序进行更新操作,系统的最终状态将是不正确的。

3 结束语

云计算是当今 IT 行业的发展方向,而作为云计算的关键技术之一,Windows Azure 具有的动态可扩展性、可用性、稳定性为部署大规模 Web 应用程序提供了可能。文中所列出的设计要点不尽全面,但是希望能为今后设计与开发类似系统提供一定的参考与借鉴价值。

参考文献:

- [1] 陈 琼. 云之战微软怎么赢[J]. IT 经理世界, 2010(24): 74-81.
- [2] McCreary D. Advanced server virtualization: VMware and Microsoft platforms in the virtual data center[M]. USA: Auerbach Publishers Inc, 2006.
- [3] 张亚勤. 微软的中国云图[N]. 科技日报, 2012-11-14(11).
- [4] Microsoft Inc. 案例研究[EB/OL]. [2014-03-09]. <http://www.windowsazure.cn/zh-cn/home/case-studies/>.
- [5] 吴彦文, 李 诗, 秦 颖. 基于微软云平台的实验学习系统的设计与实现[J]. 计算机工程与设计, 2013, 34(4): 1296-1301.
- [6] 杨静丽. 基于 Windows Azure 平台实时解决煤炭物流车辆路径问题的云服务[J]. 煤炭技术, 2013, 32(1): 273-275.
- [7] 李海勇. 基于 Windows Azure 的实验室资源平台设计与实现[D]. 上海: 上海师范大学, 2010.
- [8] 苏 凯. 基于云计算的分布式电费计算系统研究及实证[J]. 计算机技术与发展, 2013, 23(4): 217-220.
- [9] Chappell D. Introducing the Windows Azure platform[EB/OL]. (2009-12-01)[2014-03-09]. http://download.microsoft.com/download/e/4/3/e43bb484-3b52-4fa8-a9f9-ec60a32954bc/Azure_Services_Platform.pdf.
- [10] Microsoft Inc. 让云触手可及—微软云计算解决方案白皮书[EB/OL]. (2009-12-01)[2014-03-09]. <http://www.docin.com/p-41473744.html>.
- [11] 朱明中. Windows Azure 实战手册[M]. 北京: 中国水利水电出版社, 2011.
- [12] Jennings R. 云计算与 Azure 平台实战[M]. 王 鑫, 丁 斌, 译. 北京: 清华大学出版社, 2011.
- [13] Microsoft Inc. Best practices for the design of large-scale services on Windows Azure cloud services[EB/OL]. (2014-01-01)[2014-03-09]. <http://msdn.microsoft.com/en-us/library/Azure/jj717232.aspx>.
- [14] Pritchett D. BASE: an acid alternative[EB/OL]. (2008-07-28)[2014-03-09]. <http://queue.acm.org/detail.cfm?id=1394128>.

基于Windows Azure的大规模Web应用程序设计

作者：[卢朝晖](#)，[LU Zhao-hui](#)
作者单位：[海南师范大学 网络中心, 海南 海口, 571158](#)
刊名：[计算机技术与发展](#)[ISTIC](#)
英文刊名：[Computer Technology and Development](#)
年，卷(期)：2014(12)

引用本文格式：[卢朝晖](#), [LU Zhao-hui](#) [基于Windows Azure的大规模Web应用程序设计](#)[期刊论文]-[计算机技术与发展](#)
2014(12)