

云计算与云服务

信息技术的飞速发展，推动着 IT 技术的各个方面发生变革。近几年，IT 领域正加速步入云计算时代。云计算的提出引发了新的计算变革，催生出新的 IT 服务模式，进而使工作方式和商业模式发生了根本性改变。追根溯源，云计算与并行计算、分布式计算和网格计算不无关系，更是虚拟化、效用计算、SaaS、SOA 等技术混合演进的结果。通过与相关技术、商业模式、创新要素形成的有机互动，云计算已经成为信息产业实现跨越式发展的重要驱动力。

1.1 云计算的起源

对企业而言，数据中心的各种系统（包括软硬件与基础设施）是一大笔资源投入。一方面，新系统（主要指硬件部分）一般经过 3 ~ 5 年即面临逐步老化与淘汰，软件则面临不断升级的压力；另一方面，IT 的投入难以跟上业务发展的步伐，即使利用虚拟化技术，也解决不了不断增加的业务对资源的变化需求，一定时期内扩展性总是有所限制。于是，企业产生了新的需求：IT 资源能够弹性扩展、按需服务，将服务作为 IT 的核心，提升业务敏捷性，从而大幅降低成本。因此，面向服务的 IT 需求开始演化到云计算架构上。

云计算架构可以由企业自己构建，也可采用第三方云设施，但基本趋势是企业将逐步采取租用 IT 资源的方式来实现业务需要，如同使用水力、电力资源一样。计算、存储、网络将成为企业 IT 运行中可以使用的资源，无须自己建设，可按需获得。从企业角度来讲，云计算解决了 IT 资源的动态需求和最终成本问题，使得 IT 部门可以专注于服务的提供和业务运营。

现在，越来越多的企业提供的服务，不管是软件服务、信息分析、视频直播还是联机游

戏等，都通过云计算服务实现。因为云计算系统具有庞大的计算能力与存储空间，用户只要通过网络与云计算系统连接，就可以方便地取得云计算服务资源，消费者只需要按自己的使用量付费，就像付水电费那样方便、快捷。

另外，现代企业有一个关键的生产资料和要素，那就是**数据**。数据已经成为企业的核心资产，而对数据的使用和挖掘能力也日益成为企业的核心竞争力。利用云计算技术与服务能够更方便地收集、计算、挖掘数据，使那些不具备大数据基础设施的企业也能享受到大数据带来的红利。

1.2 云计算的概念

根据前面的介绍我们知道，云是一种服务，可以像使用水、电、煤那样，按需使用、灵活付费，使用者只需关注服务本身。云计算的资源是动态扩展且虚拟化的，通过互联网提供，终端用户不需要了解云中基础设施的细节，不必具有专业的云技术知识，也无须直接进行控制，只关注自身真正需要什么样的资源以及如何通过网络来获得相应的服务。

按照服务划分，云计算可以分为 IaaS、PaaS 和 SaaS 三个层次：IaaS（Infrastructure as a Service，基础架构即服务）是基础层。在这一层，通过虚拟化、动态化将 IT 基础资源（计算、网络、存储）聚合形成资源池。资源池即计算能力的集合，终端用户（企业）可以通过网络获得自己所需要的计算资源，运行自己的业务系统。这种方式使用户不必自己建设这些基础设施，而是通过付费即可使用这些资源。

在 IaaS 层之上的是 PaaS（Platform as a Service，平台即服务）层。这一层除了提供基础计算能力，还具备了业务的开发运行环境，提供包括应用代码、SDK、操作系统以及 API 在内的 IT 组件，供个人开发者和企业将相应功能模块嵌入软件或硬件，提高开发效率。对于企业或终端用户而言，这一层的服务可以为业务创新提供快速、低成本的环境。

最上层是 SaaS（Soft as a Service，软件即服务）。实际上，SaaS 在云计算概念出现之前就已经存在，它随着云计算技术的发展得到了更好的发展。SaaS 的软件是“拿来即用”的，不需要用户安装，软件升级与维护也无须终端用户参与。同时，它还是按需使用的软件，相对于传统软件购买后就无法退货具有无可比拟的优势。

对于企业而言，可根据需要使用上述某一种层次的云服务。大型企业一般需要综合上述三个层次的云服务，即层次化的云计算服务，一般也称为 I-P-S 云计算，各层可独立提供云服务，下一层的架构为上一层云计算提供支撑。例如，某视频网站采用了上述的 I-P-S 云计算架构。其中，由阿里云大型服务器群、高速网络、存储系统等组成 IaaS 架构提供基础服务；使用阿里云提供的 RDS、MQ 等服务构建在 IaaS 层上，这些 RDS、MQ 提供的服务组成 PaaS，把视频的应用逻辑（如视频编解码、视频流接入等）部署上来，提供在线的视频点播、直播等服务。这样一个大型的系统对互联网用户而言，就是一个大规模视频类 SaaS 应用。

在业内的普遍认识里，云计算按照部署模式的不同，主要分为公有云、私有云和混合云。公有云一般由云计算厂商构建，面向公众、企业提供公共服务，由云计算厂商运营；私有云是指由企业自身构建，为企业内部提供云服务；当企业既有私有云，同时又采用公有云计算服务时，这两种云之间形成一种内外数据相互流动的形态，便是混合云的模式。目前，大型企业迁云时都倾向使用混合云模式，把面向互联网用户的应用系统部署在公共云上，把内部系统或者安全等级要求较高的系统部署在私有云上。

阿里云对云的模式认识稍有不同：阿里云计算是公共服务，即公共云（Public Cloud Service），它向人们提供高科技、低门槛、简单易用的云计算服务和能力，是一种普惠服务。在阿里云公共云上，既有利用云计算检修铁路的铁路工人，也有迅速走红的直播平台小咖秀，还有 12306、中石化这样的央企……云计算为大众创业、万众创新提供了沃土。

公共云能够满足大多数客户对计算的需求，但对一些特殊的客户，公共云可能无法满足需求，比如安全方面的限制，这就要为客户提供专门的云计算服务，即“专有云”（英文为 Dedicated Cloud Service）。

阿里专有云不是另起炉灶的一套基础设施，而是公共云为满足特定客户需求而进行的延伸，是公共云的特殊形态。众多的大型企业和政府单位，也纷纷借助阿里云的技术建设了自己的专有云：中国海关总署正使用阿里云专有云上的大数据计算能力提升通关查验的效率；中国邮政集团在专有云平台上积极地打造新一代寄送业务平台；中国联通借助阿里云专有云实现了全国卡号系统的互联网云化……

与云原生（Cloud Native）的创业公司不同，传统企业内部的信息化系统已经存在 40 多年，很多内部 IT 系统部署在线下机房，包括财务、OA、HR 和报销管理等系统。随着互联网业务的蓬勃发展，更多的创新应用在互联网上涌现。企业既需要保证老系统的稳定持续运行，也需要借助云的弹性满足峰值业务访问。云下的传统系统和云上的云原生系统的有机集成，造就了混合云的形态。混合云已经成为大型传统企业进行互联网+业务创新的主要模式。阿里云既支持传统意义的混合云，也建议对于大型企业从公共云到专有云在基础设施层面遵循统一标准，确保互联互通，数据能够流动，系统能平滑迁移。只有保持底层架构一致，上层业务开发的成本才能降到最低。在这样的前提下，企业可以更方便地打通自己的专有云和公共云的通道，轻松实现混合云的架构。

因此，在本书中，我们对云服务的形态，都将统一使用“公共云”和“专有云”来进行区分和称呼。

1.3 云服务的发展现状

SaaS、IaaS、PaaS 这三种云服务模式目前处于不同的发展阶段。其中，SaaS 和 IaaS 产业比较成熟，PaaS 领域起步最晚，提供该服务的企业并不多。表 1-1 给出了三种云服务模式现状的比较。

图 1-1 给出了 2015 年中国云服务市场规模占比图，从中可以看到，国内的云计算服务以 SaaS 和 IaaS 为主，PaaS 占比较小。SaaS 服务虽然占据了半壁以上江山，但是行业格局依然比较混乱，缺乏领导者；PaaS 服务仍处于发展初期，市场认知度比较低，但是未来的增长预期恰恰是最高的；IaaS 服务的占比超过 1/3，已经得到国内企业客户的充分认可，云主机、云存储等应用形式是用户使用最多的，占比 70% 以上。

表 1-1 三种云服务模式现状的比较

	SaaS	PaaS	IaaS
服务对象	企业	开发者	开发者和企业用户
成熟度	发展最早，成熟度相对最高	起步最晚，成熟度最低	在应用层成熟后兴起，成熟度较高
核心能力	帮助企业优化业务流程	帮助开发者的产品快速获得某种功能	帮助企业 / 开发者快速拥有存储、计算等资源
发展现状	企业级市场活跃，融资集中在 C 轮及以上	潜力最大，融资集中在早期 A、B 轮	竞争激烈，垂直领域亦有发展空间
主要玩家	互联网公司、传统软件公司及创业企业	互联网公司、创业企业	电信运营商、IT 厂商、互联网公司、创业企业

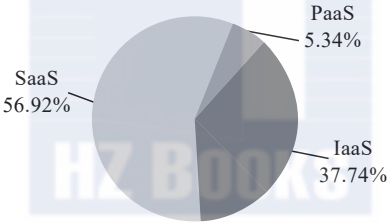


图 1-1 2015 年中国云服务市场规模占比图

接下来我们逐一分析各种云服务模式。

1.3.1 SaaS

SaaS 服务的概念产生于 2003 年，是最早引入中国的云服务形式。SaaS 是一种通过网络提供软件的模式，有了 SaaS，所有的服务都托管在云上，用户不用再购买软件，且无须对软件进行维护。现在，一个完整的企业 Web 应用程序，可以在云上提供一个敏捷、统一的企业协作平台。就像使用自来水一样，企业可以根据实际需要向 SaaS 提供商租赁在线的软件服务。SaaS 可以帮助企业减少费用，管理硬件、网络和内部 IT 部门。

早期的 SaaS 服务主要用于销售管理，如八百客、xTools 等使用的 ASP 就是早期的 SaaS 服务。进入 2008 年，传统企业集中转型，办公软件企业（如用友、金蝶）开始开发 SaaS 业务。SaaS 企业集中布局在 CRM、ERP 流程管理领域。2013 年，SaaS 服务开始细分，通信、邮箱、网盘等工具开始 SaaS 化，基于手机等移动终端的 SaaS 平台开始出现。随着

移动终端的普及，SaaS 自 2014 年以来持续高速增长。

目前，企业级 SaaS 服务市场规模保持快速增长，持续受到资本青睐，相关参与方不断探索新型服务形式。SaaS 厂商发展的基本现状可总结如下：

- 1) 传统软件企业通过拓展服务形式转型为 SaaS 服务商。
- 2) 创业公司在原有垂直领域 SaaS 服务基础上，向底层拓展同类 PaaS 服务。
- 3) 互联网巨头拓展平台，在 SaaS 基础服务上，引入第三方服务商形成 SaaS 服务入口平台，打造云服务生态。

SaaS 未来的发展趋势主要表现在三个方面：

- 1) **智能化推动流程管理效率提升：**人工智能等技术的引入将极大提升流程优化效率，改善服务，成为改进 SaaS 服务的重要推动力。
- 2) **行业垂直 SaaS 空间大：**在垂直行业积累一定客户的 SaaS 企业，有机会发掘供应链上的其他机会，例如 B2B、增值服务等。
- 3) **打通碎片化服务：**多数服务商专注单一 SaaS 流程、面临服务数据互不连通的问题，因此在提供整合碎片化数据、流程的 SaaS 服务方面存在机会。

1.3.2 PaaS

PaaS 是指将软件研发的平台作为一种服务，并提供给用户。用户或者企业可以基于 PaaS 平台快速开发自己所需要的应用和产品。同时，PaaS 平台开发的应用能更好地搭建基于 SOA 架构的企业应用。PaaS 作为一个完整的开发服务，提供了从开发工具、中间件，到数据库软件等开发者构建应用程序所需的所有开发平台的功能。

2009 年，新浪推出国内首个 PaaS 平台 SAE Alpha。2013 年，微软、亚马逊 AWS 等海外公共云 PaaS 服务进入中国；之后，BAT（百度、阿里、腾讯）分别推出开发者 PaaS 平台（百度的 BAE、阿里的 ACE、腾讯的 Qcloud）；IM、推送等领域的垂直 PaaS 平台开始提供服务（个推、融云等）。进入 2015 年，国内大型云服务商开始开放更多面向开发者的云服务能力（如阿里百川等）；同时，垂直 PaaS 平台也发展迅速，如出现了面向物流网、语音识别等的 PaaS 平台。

目前，PaaS 厂商发展的基本现状主要是在原有技术功能基础上，根据优势拓展服务形式和客户群，拓展方向主要集中在以下 3 个方面：

- **拓展服务功能：**从单一 PaaS 服务扩展多样 PaaS 服务，形成功能商店。
- **拓展服务形式：**在原有 PaaS 服务基础上，拓展同类 SaaS 服务。
- **拓展新客户：**在 PaaS 通用模块基础上，为企业提供定制化流程管理。

目前 PaaS 发展趋势主要表现在三个方面：

- 1) **业务类型同质化：**提供单一功能的 PaaS 厂商，横向发展成长为 PaaS 工具商店，可能造成 PaaS 厂商业务重合度高。
- 2) **开发者增值服务成为增长点：**在提供功能模块的基础上，形成平台生态，提供满足

企业 / 应用生命周期全流程的技术服务。

3) 国际化业务或成新增长点: 技术模块在国际化推广过程中受到地理位置、使用习惯的限制较小, 突破海外市场将成为 PaaS 厂商新增长点。

1.3.3 IaaS

IaaS 供应商为用户提供云化的 IT 基础设施, 包括处理、存储、网络和其他基本的计算资源。用户能够远程部署和运行任意软件 (包括操作系统和应用程序), 供应商则按照用户使用存储服务器、带宽、CPU 等资源数量收取服务费。

公共云 IaaS 是一种“重资产”的服务模式, 需要较大的基础设施投入和长期运营技术经验积累, 该项业务具备极强的规模效应。因此, 一旦巨头优势显现后, 将产生马太效应, 通过价格、性能和服务建立起较宽的“护城河”。亚马逊自 2006 年推出 AWS 服务, 到 2016 年 4 月, 其价格已经下调了 42 次; 阿里云从 2015 年 10 月到 2016 年 10 月, 一年内降价高达 17 次。

IaaS 的发展大致可以分为萌芽阶段、IaaS 成长阶段和洗牌阶段。

❑ IaaS 萌芽阶段: 2008 年, IBM 在中国建立首个云计算中心, IaaS 正式进入中国。随后一年, 盛大、阿里云开始研发和试点运营相关云业务。

❑ IaaS 成长阶段: 2013 年前后, 微软、亚马逊 AWS IaaS 业务正式进入中国。同年, UCloud、青云等 IaaS 创业公司成立并开始提供服务, 腾讯、华为等巨头也纷纷加入云服务阵营。

❑ 洗牌阶段: 到 2015 年, IaaS 云行业发展趋于稳定, 行业格局和盈利模式日渐清晰, 行业领先者开始出现, 企业逐渐开始盈利。同时, 行业竞争加剧, 进入洗牌阶段。

2017 年 1 月, 阿里巴巴集团财报显示, 阿里云 2017 财年第四财季 (2017 年 1 月 1 日至 2017 年 3 月 31 日) 营收达到 21.63 亿元, 同比增长 103%, 连续第八个季度增幅超三位数。阿里云云计算付费用户数量同比增长 100%, 达到 87.4 万, 覆盖金融、医疗、公共交通、能源、制造、政府机构、游戏、多媒体等广泛的行业和企业类型。

1.4 云计算的未来

云服务正在快速增长, 它越来越广泛地被更多客户接受和使用, 成千上万的企业愿意把业务系统部署在云上, 上云 / 迁云已经成为企业 IT 的大趋势。不仅是企业, 越来越多的普通民众也切实感知到云计算和大数据给日常工作、生活带来的变化, 人们都在不约而同地问: “云计算的未来在哪里? 云计算将带我们走向何方?” 阿里云与大家一起探求这个问题的答案, 作为每天都在与云打交道的从业人员, 我们也希望与读者分享作者团队的见解。

首先, 云计算会更智能。谈到这个话题, 一定离不开大数据, 大数据和云计算息息相关,

被认为是硬币的两面。数据爆发式增长和分布式计算能力的提升，让我们拥有了大数据的分析和计算能力，通过专家经验、行业模型和机器学习，结合大量的数据获得了一部分洞察，然后基于洞察进行预测。但是，这只是在人类在 DT (Data Technology) 时代迈出的第一步。

当面对更大的数据量、更复杂的场景时，我们需要云计算提供更强的计算能力、更大的存储能力以及更好的算法、模型和计算平台，甚至希望减少人的经验在模型训练中参与度，这是大数据走向人工智能的要求。此时，由神经算法网络衍生出的深度学习可以基于大量数据进行自我学习，在图像、语言等富媒体分类和识别上取得良好的效果。谷歌人工智能产品——围棋机器人 AlphaGo 在击败了世界冠军、职业九段选手李世石后，注册了 ID 为“master”的帐号，在中国棋类网站连续对决中日韩数十位高手，连续 60 局无一败绩，甚至在 2017 年 5 月直落 3 盘完败世界排名第一的中国选手柯洁。人人都惊呼：“奇点临近了，人工智能的时代到来了！”AlphaGo 的核心就是神经网络和深度学习算法。

阿里巴巴将自己定位为大数据公司，首先是因为我们拥有众多高质量的数据；其次，阿里云已经发布的大数据平台“数加”，覆盖数据采集、计算引擎、数据加工、数据分析、机器学习、数据应用等数据生产全链条。同时，与其他云厂商的技术不同，阿里云在人工智能基础上加入大数据，形成了数据智能。阿里云的人工智能机器人 ET 正是人工智能和大数据结合的产物，ET 背后使用了多项人工智能的技术，涵盖自然语言识别、实时视频分析、语音识别及语音合成、人机交互、知识图谱等。

新技术的产生会给云计算带来颠覆性的改变，这些新技术很大程度上将来自于物联网、量子计算、大数据和人工智能等领域。

其次，云计算会更安全。在云计算概念出现的早期，数据泄露和数据丢失等安全问题是最受关注的话题。数据作为企业的核心资产，要保证不丢失、不被偷窥和窃取，甚至要防止云计算厂商监守自盗。今天，大量的企业已经迁移到云上，甚至一些创业公司从创建第一天起就已经在云上。更多企业上云促使更多的云上安全问题暴露，如不安全的接口、DDoS 攻击、内部员工的恶意破坏等。

用户数据安全是阿里云商业的前提，用户的业务永续是阿里云产品的座右铭。为此，2015 年 7 月，阿里云发布《数据保护倡议书》并承诺：“绝不碰客户数据”。同时，阿里云的产品开发、发布、运维和变更的流程均将可用性作为核心的指标进行考核。此外，阿里云开发了专用的安全产品“云盾”，其中集成了阿里巴巴集团多年安防经验，在线为成千上万的客户保驾护航。特别值得一提的是，阿里云也将大数据能力植入到安全体系中，比如研发了“态势感知”这款产品。阿里云的计算集群每天会调用超过数十万个 CPU 核心用于海量数据的计算，从而分析每一个客户遇到的安全问题，这些数据来自于客户在开通态势感知时授权云盾出于安全的目的进行计算。阿里云在安全态势感知上的挑战，是替整个安全行业在探索大数据的边界和瓶颈。在 2017 年，云盾的产品会像态势感知一样全面拥抱数据化。

最终，云计算会使生活更美好。得益于云计算，我们的生产和生活开始变得美好：新浪微博在每年春节只需要临时租用云上虚拟服务器，就可以轻松应对成倍的业务压力，这为他

们节省了上千台服务器的采购费用；阿里云在杭州实施的“城市大脑”，通过智能调节红绿灯让车辆通行速度最高提升了 11%；阿里云 ET 能够挑战人类的优秀速记员为会议提供速记服务，甚至还能解说体育比赛；支付宝可以用刷脸进行支付……

但是，我们看到，云计算依然有很多需要完善和提高的地方：今天，云计算服务还无法像苹果手机那样实现任何普通人都能轻松地操作；云计算的资源也无法像水、电、煤气那样作为一种公共基础设施而自由使用；云计算中还有许多技术难题等着技术人员去攻克。

云计算是一个技术概念，同时也是一个商业概念，但云计算的本质是计算，正是由于不断地计算，才使得人们一次次突破科技瓶颈，推动科技进步，促进人类文明发展。计算无处不在，生活中处处需要计算。计算的终极意义是发挥数字的力量，解决问题、创造价值，让数字不止于数字，赋予数字以人的喜怒哀乐。

阿里云不满足于仅仅成为一家科技公司，我们一直致力于推动计算成为普惠科技，成为各行各业能够简单获取的能力，进而不断去创造“无法计算的价值”。在科技落后的过去，“未来”往往是一个只存在于想象的憧憬却永远不会在有生之年实现的概念。今天，云计算给人类插上了想象的翅膀，让我们飞上云端，让更多的不可能成为可能，让我们所谈论“未来”变成看得见、摸得着的更美好的明天。

第 2 章 Chapter 2

企业迁云概述

企业使用 IT 架构的历史，从最初大型机的集中式计算架构到以服务方式提供计算能力的云架构，走过了几十年的历程，每次 IT 系统升级的变革，都带动着生产力和企业管理的变革与发展。图 2-1 给出了企业 IT 系统的发展历史。

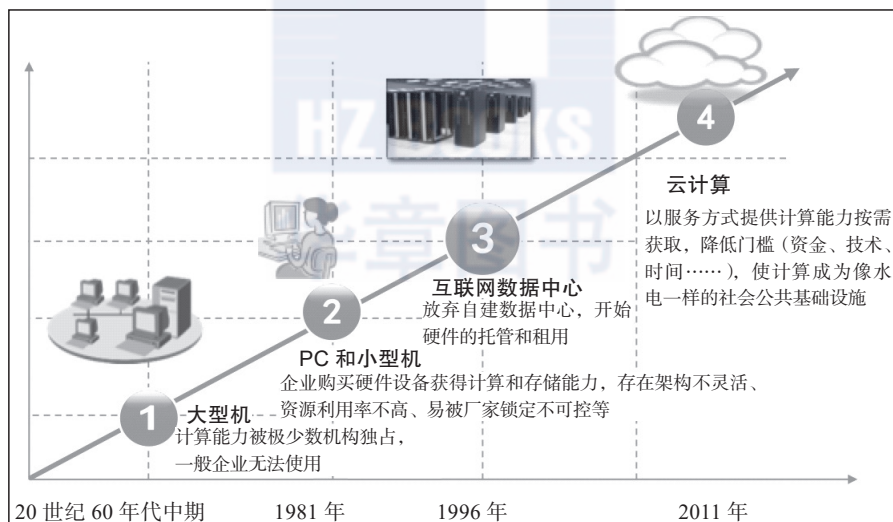


图 2-1 企业 IT 系统的发展历史

1) 从 20 世纪 60 年代中期开始，企业开始使用大型机。大型机的使用门槛和成本很高，只有极少数企业能够使用，计算能力被极少数企业占有，而一般企业无法使用这类资源。

2) 从 20 世纪 80 年代开始，PC 和小型机出现，企业通过购买硬件获得计算和存储能力，但存在架构不灵活、资源利用率不高、易被厂家锁定而不可控等问题。

3) 从 20 世纪 90 年代中期开始, 企业开始在互联网数据中心托管和租用硬件, 放弃自建数据中心。风、火、水、电由数据中心提供和保障, 但是计算设备以及计算系统还需要企业自身提供, 企业的 IT 运营成本并没有明显的下降。计算能力依然被少数大企业占有, 一般企业由于资金、技术等因素无法享受到计算带来的便利。

现在已经进入云架构时代, 以服务方式提供计算能力, 按需获取, 使用门槛降低, 减少了企业在资金、技术、时间等方面的投入, 使计算成为像水电一样的社会公共基础设施。

云架构的弹性带来资源的集约化, 实现按需使用、按需付费以及高质量的专业运维带来计算的服务化。通过 IT 集中式架构向分布式云架构的转型优化, 能够支撑高并发、高性能的架构需求, 使企业能够放心地拥抱互联网, 能够进行基于互联网、大数据的业务创新。

人类已经从 IT 时代走向 DT 时代, IT 时代是以自我控制、自我管理为主, 而 DT 时代是以服务大众、激发生产力为主。这两者之间看起来似乎是技术上的差异, 但实际上是思想观念层面的差异。

对于企业 CIO 而言, DT 时代, 他们在思想观念上有哪些方面需要提升呢? 笔者认为, 至少在企业的 IT 架构方面, 他们需要做到以下考虑:

- 1) 建立一个敏捷的 IT 架构, 以满足多变的业务需求。
- 2) 提供充分的扩展性和灵活性, 应对业务的快速创新。
- 3) 优化成本、资源, 提高业务运营效率。
- 4) 技术和业务的无缝融合, 通过容器化、微服务、Devops 等先进工具、技术, 加快技术迭代周期, 降低业务试错成本, 用技术拓展商业边界。

企业上云能很好地满足企业 CIO 的诉求, 体现在如下几个方面:

- ❑ 显著降低 TCO (Total Cost of Ownership, 总所有成本), 无须投入购买大量硬件和软件的费用, 减少人员成本, 降低运维费用。
- ❑ 提供系统稳定性, 无须关注底层基础设施的运维, 云计算提供商可提供专业的运维能力。
- ❑ IT 资源可弹性提供, 实现资源按需购买、灵活调度以及方便系统的扩容和缩容。
- ❑ 实现数据的集中, 企业可以存储更多的数据, 方便数据共享, 让沉积的数据产生价值。
- ❑ 架构可复用, 统筹规划, 统一建设, 可减少代码的重复开发, 提高研发效率, 缩短业务开发周期。架构的可复用也有利于系统的规模化运维。
- ❑ 企业数据、IT 架构整合提高系统能力以及更好地支持业务创新, 整合主要体现在数据整合、应用整合、SOA 等方面。很多企业在过去几年中或多或少在这几个方面进行了有益的尝试。现在, 以企业云化为契机, 进行企业数据和架构的整合是一个不错的选择。

正是因为具有上述诸多的优势, 企业的 IT 管理人员正在越来越广泛地接受云计算这一新生事物, 互联网行业最天然地拥抱了云计算, 而传统行业的 IT 决策者们也纷纷开始试水云计算。那么, 对于 CIO 和 CTO 们来说, 企业迁云后, IT 管理将会发生哪些变化呢?

2.1 企业迁云后 IT 管理的改变

企业迁云后，原有的 IT 管理模式会得到极大的改变，具体表现在如下三个方面：

(1) 企业 IT 投入模式改变

在传统的 IT 投入模式下，基础设施（机房、电力、带宽、主机、存储、网络、数据库、中间件）的投入占比最大，服务器运维人工成本次之，应用开发的投入占比最少。上云后，投入比例正好相反，基础设施租用费用最少，运维的人工成本也将大量下降，这样可以把更多的资金投入业务开发上面，企业可以快速满足业务的需求。企业迁云后投入的变化如图 2-2 所示。

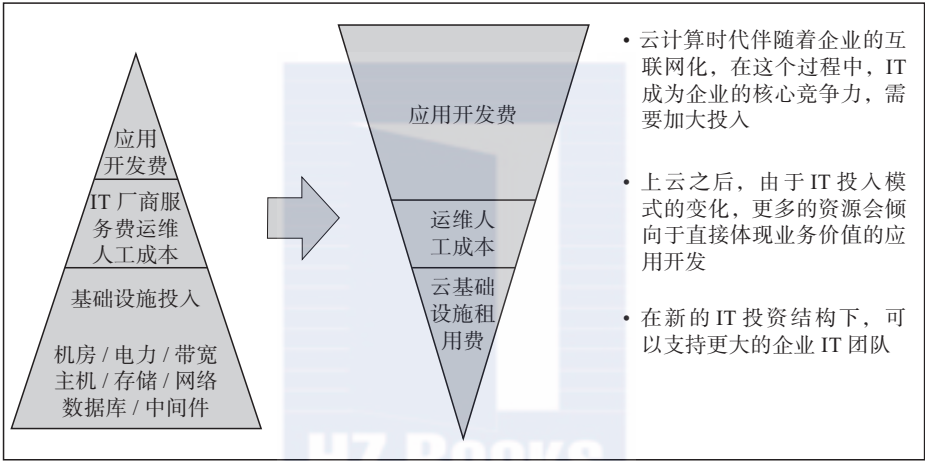


图 2-2 企业上云后投入的变化

(2) IT 人员能力要求的改变

企业迁云后，对 IT 人员能力的要求也将发生变化，如表 2-1 所示。

表 2-1 企业迁云后 IT 人员能力要求的改变

原有的模式	新的模式
被动地理解业务需求	主动发现客户需求
解决方案完全依靠 SI/SV	探索业务模式，定义解决方案技术架构
偏重与招标采购，项目管理	偏重对客户理解，对新技术的掌握
按项目进行验收，后期运维	业务不停，改进不止
采购员 + 监工	产品经理 + 架构师

迁云前，IT 人员更关注底层基础设施的运维、项目管理，对业务需求的理解和响应比较被动。迁云后，IT 人员需要更关注业务的需求，探索新的业务模式，主动发现客户需求，寻找新的技术和解决方案，而不必关注底层平台的运维和资源管理。比如，以往 IT 人员遇到一个系统的问题，在解决问题的同时往往还要研究清楚问题发生的根本原因是什么，以避

免下次发生同样的问题。在云计算的环境下，这种思路也会发生改变，我们需要迅速下线或释放有问题的资源，同时开通相对应的新资源。所以，企业 IT 人员需要做的是准备好问题发生时的预案并及时执行，至于这个问题是因何产生的，则可以留给云计算厂商的工程师去思考和研究。

(3) 整个企业 IT 运营模式的改变

迁云后，整个企业的 IT 运营模式将发生改变，如图 2-3 所示。

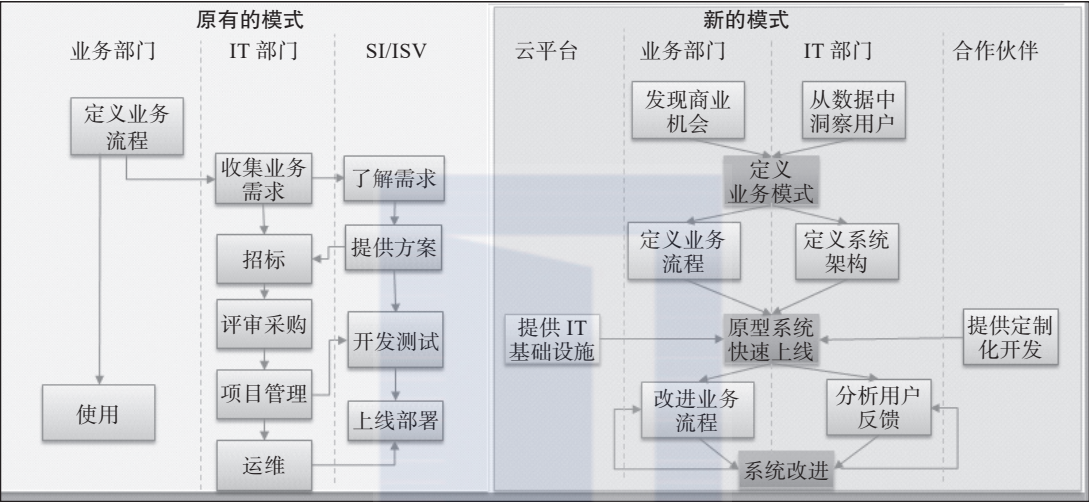


图 2-3 企业迁云前后运营模式的改变

在原有的 IT 管理模式，由业务部门定义业务流程，IT 部门收集业务需求，要通过招标、评审采购、项目管理等流程，同时同步给软件开发商，开发测试完成后，还要经历业务上线到后期转运维等一系列流程，时间跨度很长。迁云后，业务部门和 IT 部门通过各自的角度定义业务模式，转入流程设计和系统架构设计，通过云平台提供的 IT 基础设施，由负责软件开发的合作伙伴迅速转入定制开发。企业后续无须对系统进行大量运维，无论是业务部门还是 IT 部门都更关注业务流程和用户体验，双方共同促进对系统持续的改进。

从上述介绍可以看出，企业迁云会带来投入、管理、运营模式上的极大变化，大大节约了企业的资金和人力成本，使企业能将注意力集中在核心业务方面，从而创造更多的价值。

2.2 企业迁云需要注意的问题

云计算作为企业向互联网+转型的核心基础服务，正以前所未有的速度扩张。事实上，就 RightScale 近期公布的云报告显示，93% 的企业已采用了一些云计算的能力。那么，对于一家业务稳定的公司，若被云计算“弹性、灵活、安全、低成本”的优势所吸引，有计划迁移到云端，那么在正式启动迁云工作之前，需要明确以下几个问题，使迁云之路更加清晰、

顺畅。

(1) 企业是否真的需要迁云

不可否认,云计算有很多优势,能为企业带来更多便利,比如省钱、省力,让业务更有弹性、更灵活,可以便捷地获取大数据分析能力。但我们也要清醒地认识到,云计算不是万能的,因此不是每一家企业都需要迁云或者全面迁云。在做出迁云的决策前,请先思考如下5个问题,以确定企业是否真的需要迁云:

- 1) 企业的用户是否分布在全国各地,地域跨度较大?
- 2) 企业是否已在基础IT环节投入了较大的成本但运行效果并不满意?
- 3) 企业的业务是否在快速发展,或者即将进入爆发期?
- 4) 企业是否遇到或可能遇到安全问题,以致影响企业公共业务的运行?
- 5) 企业的竞争对手是否已迁云并取得了较好的效果?

如果上述5个问题中能确定其中一个是肯定的答案,那就可以着手选购云计算服务;否则,就需要再次权衡是否值得付出迁云的成本与风险。

(2) 选择并确定最优的云架构方案

在明确了企业确实需要迁云之后,接下来就要根据企业的实际情况选择一种最佳的云架构方案。

1) 选择混合云\公共云。如果企业拥有大量的自购IT基础设施,有自建或租赁的机房,或有一些敏感数据仍需保留在自己的物理服务器上,则可以通过使用虚拟专用网(VPC)来构建混合云,作为阶段性的过渡方案。如果企业尚未搭建IT基础设施,那么公共云是最佳的选择。

2) 设计系统架构

在确定了选择混合云或公共云之后,接下来就要设计系统架构,一般有以下几种方式:

- ❑ 一般企业会根据在传统IDC搭建IT系统的方式,先做出一个方案,作为上云的基础和比较方案。其中主要应考虑资源的规模、性能要求、扩展性要求等。
- ❑ 在一些较大的云计算平台(如阿里云)上,会提供多媒体、医疗、游戏、物联网、游戏等多个行业的解决方案,有丰富的典型部署案例可供参考。企业可以照葫芦画瓢,借鉴同行的经验做出自己的方案,并通过服务价格计算器来预估整体预算。
- ❑ 如果自己没有把握,可以通过售前技术服务进行咨询,大部分云计算平台均提供此类服务。

(3) 前期测试

确定系统架构后,接下来可根据业务的部署方案选取相应的云产品及服务进行功能、性能和安全性测试。

在阿里云上,所有产品均采用自助式服务,用户只需注册阿里云账号、开通相应的测试服务实例即可。同时,阿里云还提供4款核心云产品的免费试用。通过免费试用,用户可以顺利确定适合自己企业的迁云方案。

（4）部署迁移

根据测试结果，用户可以购买需要的产品与配置，开始迁移上云。后面各章将详细介绍部署迁移的过程，这是本书的核心内容，这里不再赘述。

在上述过程中，还有几个注意事项不可忽视。首先，各大云平台会不定期推出一些优惠政策，特别是针对新购客户。例如，阿里云会经常推出企业购买云服务器 ECS 和云数据库 RDS 的优惠策略。本着能省就省的原则，建议用户关注这些优惠政策。其次，如果是在云端构建新业务，则需要先完成域名备案。最后，遇到重要活动或事件，很可能需要进行服务保障，这种情况下应提前通知云服务提供方做好准备。以阿里云为例，会提供专业的护航服务，以确保帮助企业用户度过类似游戏公测上线、红包秒杀、双十一促销这样的流量洪峰。

2.3 阿里云服务简介

阿里云于 2009 年正式成立，现已成为国内领先的云服务提供商。阿里云在全球各地部署高效节能的绿色数据中心，利用清洁计算支持不同的互联网应用。目前，阿里云已在杭州、北京、青岛、深圳、上海、千岛湖、内蒙古、中国香港、新加坡、美国硅谷、俄罗斯、日本，以及欧洲、中东等地设立了数据中心，未来还将设立更多新的数据中心。

阿里云目前已有上百款产品，可满足用户在云计算、大数据方面的需求，产品包括域名、备案、解析、计算、存储、网络、数据库、安全、数据分析及互联网中间件。尤其是互联网中间件产品来源于阿里巴巴多年来在互联网应用实践方面的经验积累，非常适合基于互联网平台的电商、金融等企业的业务特点，包含了整套成熟的分布式计算框架（包括分布式服务化框架、服务治理、运维管控、链路追踪和稳定性组件等）。目前，这些产品已经被多家互联网金融客户使用，帮助这些企业级客户轻松构建并托管分布式应用服务体系。针对外部黑客攻击，阿里云有数百名安全相关的技术和运营人员，构建了完备的云盾安全体系，通过大数据分析，实时掌握互联网上攻击的最新情报，并根据这些情报制订防护策略，实时保护客户在阿里云上的资源。图 2-4 给出了截至 2017 年 4 月阿里云的产品分类框架。

阿里云云平台为五层架构（见图 2-5），所有云产品都是构建在第三层，即飞天操作系统之上，通过飞天操作系统底层的任务调度、文件系统、远程过程调用、资源管理等实现各种云产品服务。

- 第一层：基础设施层，包括数据中心的风、火、水、电以及服务器、网络等设施。
- 第二层：大规模 x86 服务器上部署的单机操作系统层，操作系统为阿里云版 Linux。
- 第三层：阿里云平台飞天操作系统层，它是整个云平台的核心层，包括集群部署、集群监控、分布式文件系统、安全管理、分布式协同服务、远程过程调用、资源管理和任务调度八大模块。
- 第四层：阿里云产品服务层。用户通过这层实现云产品的使用，包括弹性计算、数

数据库与存储服务、大数据类的数加产品服务、应用服务等。

□ 第五层：该层是各种应用系统基于阿里云云产品开发可架构，实现对外提供服务的
应用层。



图 2-4 阿里云产品的框架

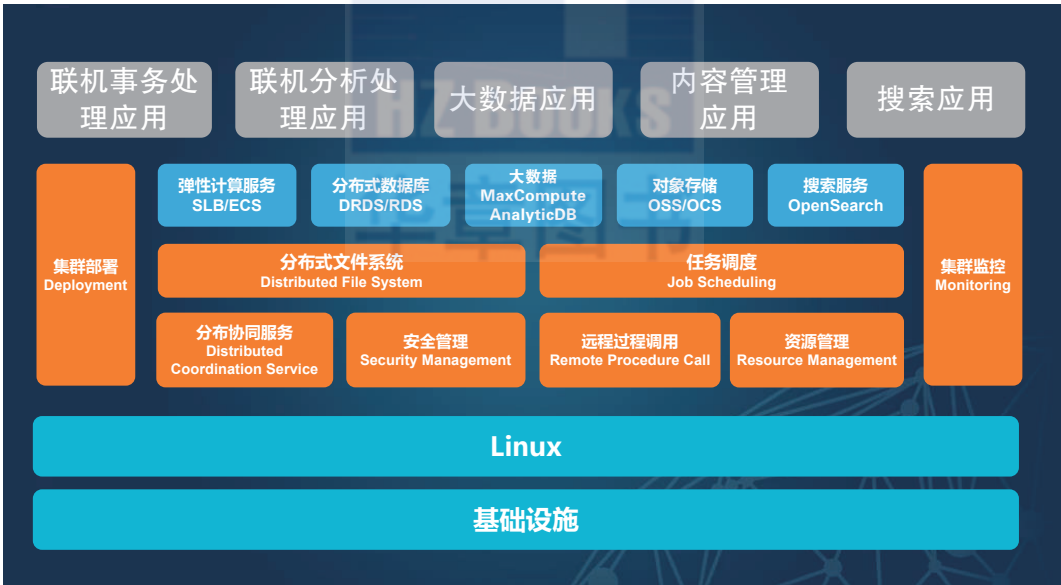


图 2-5 阿里云云平台架构

阿里云云平台通过这种五层平台架构，实现了超大规模的集群运算能力，使用户更方便、高效地使用云计算服务。

2.4 云端实践

事实上，阿里云已经深入到人们日常生活的各个角落，每天我们从网络上获取的各类服务后端都可能使用的是阿里云计算服务。下面将通过三个例子简要介绍阿里云的实践。

2.4.1 12306 网站

高峰时期的 12306 网站可以说是世界上最繁忙的网站，它与公众的日常生活密切相关。2015 年春运售票最高峰日出现在 2014 年 12 月 19 日，网站访问量（PV 值）达到破纪录的 297 亿次，平均每秒 PV 超过 30 万次，当天共发售火车票 956.4 万张，其中互联网发售 563.9 万张，占比 59%，创历年春运新高。12306 网站顶住了大并发请求的“集中轰炸”，其中很重要的一个因素是采用了阿里云的技术。

12306 网站在迁云之前主要面临以下三个方面的问题：

1) 火车票查询业务占 12306 网站全部流量的 90% 以上，业务高峰时期请求非常密集。查询的性能要求是业务系统中最重要的一环，也是往年造成网站拥堵的主要原因之一，如何支撑住峰值流量显得无比重要。

2) 12306 网站对于安全防护要求很高，对互联网各类工具流量的分析和识别要求非常高。

3) 系统可用性要求极高，必须 7×24 小时不间断服务，没有非计划性宕机时间。

通过利用阿里云的混合云方式，12306 网站顺利解决了高并发、大流量的问题：

1) 12306 把余票查询系统从后台分离出来，在“云上”独立部署了一套余票查询系统，通过阿里云平台来支撑余票查询环节的访问量（占 12306 网站近 90% 的流量），并根据系统压力情况随时动态扩容服务器，解决了往年峰值流量造成的网站拥堵问题。

2) 使用混合云架构，通过阿里云对 12306 网站自有机房的容量提供有效扩展，余票查询系统做到了按需获取所需要的服务器资源，并可以动态调整网络带宽，利用这些可扩展的资源解决了高流量和高负载情况下系统因无法快速弹性扩展而导致的性能瓶颈和系统崩溃问题。

3) 多数据中心的混合云模式提高了 12306 网站的灾备能力，云上云下互为灾备，极大提高了业务的持续服务能力。

2.4.2 淘宝双十一活动

每年的淘宝双十一活动都是电商和消费者的狂欢日，但都给淘宝网站带来巨大的考验。双十一的活动很多，包括秒杀、红包、直播等，且业务开发更新频繁，同时对稳定性要求很高，必须保证系统时刻稳定可靠。总的来讲，双十一活动主要存在如下特点：

- ❑ 高并发性能仍然存在瓶颈。
- ❑ 扩展能力对大系统不很灵活。
- ❑ 各垂直系统相同模块依然重复。
- ❑ 各研发部门各自开发相同和相似的功能模块，无法沉淀成阿里统一的应用服务。

- ❑ 新功能版本开发、更新仍然存在困难。
- ❑ 基础设施成本很高。

随着云计算的发展，淘宝逐渐开始迁云，到 2016 年双十一，90% 的业务流量都在阿里云上运行。基于阿里云的互联网企业级架构 PaaS 和底层基础设施 IaaS，淘宝已构造出世界级高可用、高并发的网站，如图 2-6 所示。



图 2-6 淘宝基于阿里云平台的架构

淘宝通过使用阿里云提供的 EDAS（应用分布式框架）、ONS（消息中间件）、DRDS（分布式数据）构成的高性能应用分布式集群框架，构建统一的共享服务层，通过对业务进行服务化的改造，拆分成用户中心、商品中心、交易中心、评价中心等多个共享服务中心，各中心保证服务稳定性、可靠性。淘宝、天猫的业务研发部门要实现如用户登录、商品管理等，只需要调用共享服务层的服务即可。服务调用的链路监控和管理由应用分布式服务统一管控和展现。基础设施使用阿里云的 IaaS（基础设施即服务）实现底层资源的弹性。应用中间件通过使用应用分布式框架解决方案，使淘宝在交易订单、商品库存、支付等方面做到了服务化、异步化，实现应用架构的弹性。这样，淘宝整个系统实现了线性无限扩展、海量并发，并且任意节点链路都可以保证高可靠性。应用共享服务中心的建立，也使部门数据能共享打通，数据被自然沉淀，并且系统新版本开发更新更敏捷，业务创新速度更快。另外，大量弹性资源的使用，使资源成本显著降低。

2.4.3 视频点播 / 直播类网站

4G 的普及催生了微视频，促进了游戏视频、移动视频业务的发展，新一波创业潮催生

的初创公司以及传统行业互联网转型的自媒体、新媒体、播客的发展，这些都进一步推动了视频点播 / 直播类系统的发展。而该类系统对计算、存储以及带宽有很高的要求。

(1) 视频点播类系统的特点

视频点播类系统有如下特点：

- 1) 下行流量少，上行流量多，存播比约为 1 比几十、几百甚至成千上万。
- 2) 对上行带宽要求极大，对 CDN 要求极大，遇到业务高峰时将对系统产生更大压力。
- 3) 对存储要求极大，多路转码需要分别存储。
- 4) 用户数据需要永久保存，有归档需求。
- 5) 对转码效率要求高，多为离线转码，宽屏影视客户的转码需求更加苛刻。

通过阿里云视频开放平台、CDN 视频加速服务、海量弹性带宽和存储可向客户提供整套解决方案，如图 2-7 所示。

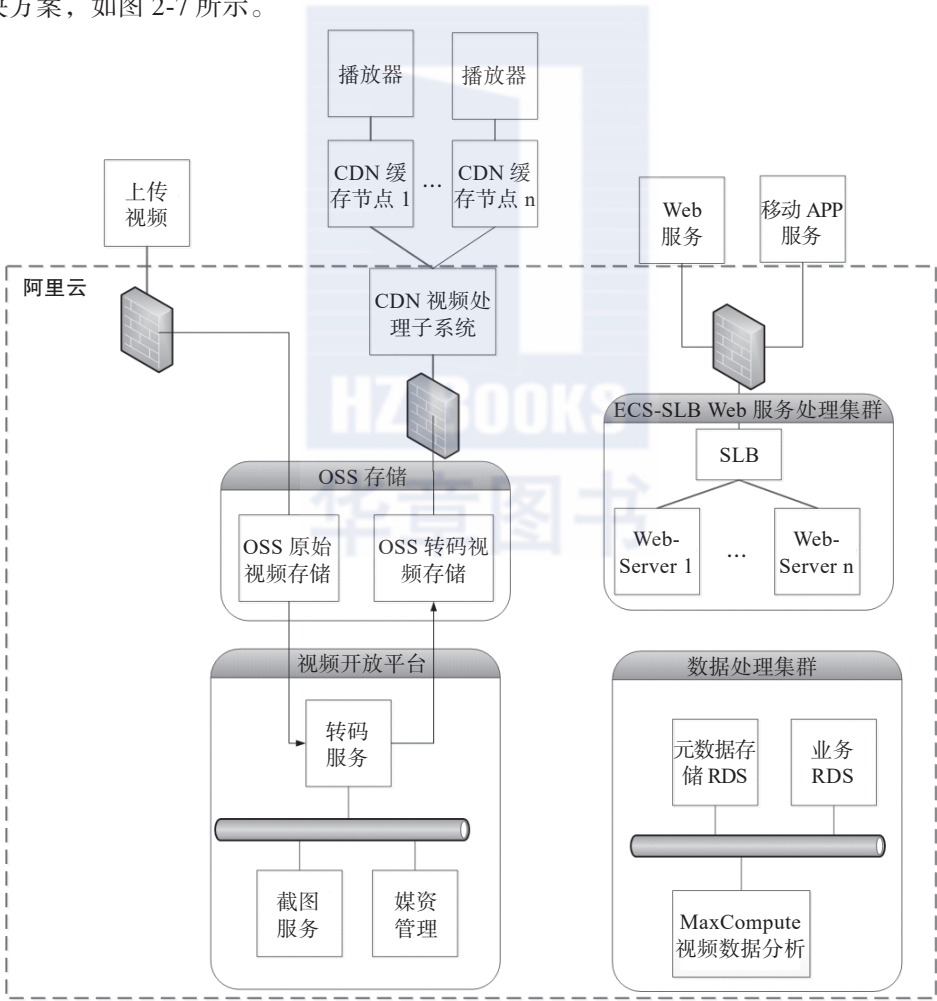


图 2-7 基于阿里云的视频点播类网站架构

- 阿里云视频直播解决方案的优势主要在于：
- 1) 阿里云的视频开放平台、CDN 流媒体加速，视频播放 PaaS 平台所提供的转码、图片处理、媒资管理等功能可降低平台开发难度，从而能快速搭建视频播放平台，并且 CDN 流媒体加速能将海量流量压力分散到全国，通过切片加速方式支持视频内容的时移播放。
 - 2) 使用 OSS 产品按需存储，OSS 开放云存储支持多路视频转码的存储。
 - 3) 使用 ECS 云服务器搭建自媒体视频播放平台实现系统弹性扩展。
 - 4) 使用 MaxCompute（原 ODPS）大数据分析平台，对用户观影行为进行分析，进而进行精准营销，推送用户可能感兴趣的视频资源。

(2) 直播类业务

直播类业务的特点与点播类业务类似。直播类业务的驱动力来自于 4G 催生的赛事等内容直播以及传统媒体向新媒体的转型。系统架构方案上，同样以转码和 CDN 为核心，但转码为实时转码，对运算效率要求更高，CDN 为实时加速。主要解决流媒体视频直播功能，需要将接收的 H.264/AAC 等流媒体码流通过实时转码技术转换成 HLS、RTMP 等协议模式，输出流畅、高清、标清等码流以适配不同终端。同时，为支持视频存档，所有视频在直播的同时还要在 OSS、OAS 长期保存以备留档、点播。

- 而现在的直播类业务往往有实时交互的功能需求，系统主要存在如下特点：
- 1) 通信不再是单向为主，而是双向互动，要求即时性强，系统延迟尽量少。
 - 2) CDN 加速无法完全解决这类需求，需要靠就近部署服务器解决延迟问题。
 - 3) 对带宽要求极大，不论是源站还是 CDN 节点。
 - 4) 业务直播视频对实时转码要求很大。
 - 5) 对稳定性要求极强，某地区服务器宕机会影响该地区所有业务。

基于阿里云的直播网站架构如图 2-8 所示。

业务迁移至阿里云后，通过阿里云遍布世界各地的数据中心，利用阿里云 CDN 网络，可以实现视频用户的就近访问，从而减少用户的访问延时。阿里云数据中心网络的多线路接入，能够保证使用不同运营商用户访问云端视频服务，链路

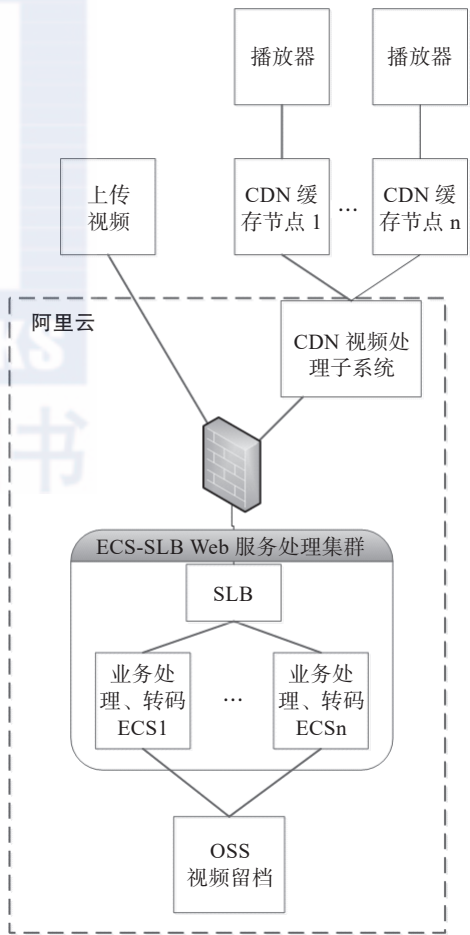
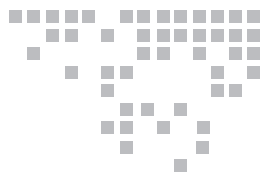


图 2-8 基于阿里云的直播类网站架构

更稳定、延时更少。阿里云的带宽资源可弹性购买，从而解决客户对大带宽购买需求。图 2-8 中，SLB 可以把带宽流量均匀转发到后端 ECS 服务，实现用户访问的良好体验；通过 OSS 的存储，能方便地进行视频留档；CDN 视频处理子系统能处理直播视频的在线转码，从而大大简化视频直播开发的复杂度。上云后，能通过业务转码、视频存储、带宽访问等提供一整套视频直播与实时交换解决方案，使企业更迅速地实现业务开发需求，降低开发成本，在激烈的市场竞争中抢占先机。





云上通用架构设计与改造

本章将介绍如何基于常用的阿里云产品和技术，快速构建出满足客户业务需求的架构。在进行实际架构设计和实施过程中，将就如何恰当地使用阿里云产品与技术给出建议和指导，同时会对阿里云产品背后的技术细节进行探讨。

3.1 网络架构设计

网络架构设计是在云上进行业务技术架构设计的第一步，本节将结合阿里云基础网络产品介绍如何进行网络架构设计、规划和建设。

3.1.1 网络产品简介

进行网络架构设计时离不开网络产品的应用，所以我们先初步了解一下构建网络时需要用到的网络产品。

1. VPC

经典网络和专有网络是阿里云上的两种网络形态。专有网络（Virtual Private Cloud，VPC）是阿里云近年推出的一种新的网络形态，可帮助用户基于阿里云构建出一个隔离的网络环境，因此我们通常建议新用户直接使用专有网络，不要再使用经典网络。通过 VPC，阿里云上的用户可以完全掌控的自己的网络环境，包括选择自有 IP 地址范围、划分网段、配置路由表和网关等。此外，也可以通过专线 /VPN 等连接方式将 VPC 与用户其他数据中心组成一个按需定制的混合型网络环境，实现阿里云和其他数据中心的互联互通。

2. 高速通道

高速通道（Express Connect）是一款基于 IPVPN 的高效网络服务，用于在阿里云上的不同网络环境间，如多个 VPC 之间、客户自建 IDC 机房和阿里云 VPC 之间，实现高速、稳定、安全的私网通信，可有效地帮助用户提高网络拓扑的灵活性和跨网络通信的质量和安全性。

对于多个 VPC 之间的内网通信，可实现：

- 1) 无论它们位于相同地域还是不同地域，都能实现远距离数据快捷传输。
- 2) 无论它们归属于同一账号还是不同账号，使合作开发灵活方便，提升开发效率。

对于物理 IDC 和阿里云上 VPC 之间的内网通信，实现了融合网络。

3. VPN

VPN，即虚拟专用网络，可用于在公用网络上建立专用网络，进行加密通信。VPN 网关通过对数据包的加密和数据包目标地址的转换实现远程访问。VPN 有多种分类方式，常用的是按协议进行分类。VPN 可通过服务器、硬件、软件等多种方式实现。阿里云上的 VPN 网关需要用户基于阿里云 ECS 服务器和 VPN 软件实现。

3.1.2 网络接入与规划

在阿里云上进行网络架构设计与规划时，客户首先需要考虑云上网络环境是否需要和传统数据中心互通，以及云上云下 IP 地址划分、网段规划等工作。

1. 高速通道接入

阿里云公共云上如果需要使用高速通道接入，客户必须使用 VPC 专有网络环境。目前高速通道主要支持用户自建数据中心或将其他云计算平台接入阿里云以及阿里云 VPC 网络内部不同地域或不同账号之间的网络接入场景。

通过高速通道接入阿里云的准备工作如下：

1) 明确专线条数。在具有一条以上专线时，阿里云专线接入冗余方式默认采用主备方式，通过管理距离来定义线路优先级。

2) 明确专线带宽速率。一般有 2M/4M/8M/10M/100M/1G/10Gb 几种规格，速率越大，带宽越高，成本越高。

3) 明确专线接入类型。目前常用的专线类型主要有光纤和以太网两种，阿里云都可以支持。建议客户使用光纤接入。

4) 明确业务需求，确定 VPC 的预期规模。在创建 VPC 时可根据规模来选择网段类型。阿里云 VPC 有三类可选子网：① A 类地址，网段为 10.0.0.0 ~ 10.255.255.255 (10.0.0/8)，默认子网掩码 255.0.0.0，最多可以容纳 1677 万多台电脑；② B 类地址，网段为

172.16.0.0 ~ 172.31.0.0 (172.16.0.0/12), 默认子网掩码 255.224.0.0, 最多可以容纳 6 万台电脑; ③ C 类地址, 网段为 192.168.0.0 ~ 192.168.255.255 (192.168.0.0/16), 默认子网掩码 255.255.255.0, 最多可以容纳 254 台电脑。

基于阿里云高速通道接入的步骤如下:

- 1) 填写并提交相关资料信息, 等待阿里云审核, 一般 1 个工作日左右完成。
- 2) 完成审核后再登录高速通道控制台, 选择对应地域申请专线接入。
- 3) 审批通过后进行高速通道接入测试。

如果是自建数据中心或其他云计算平台接入到阿里云 VPC 网络, 则需要选择物理专线。物理专线接入需要用户自行联系运营商实施, 阿里云负责配合用户物理专线接入的调试工作。^①

如果是在阿里云 VPC 网络环境内部不同 VPC 网络间的连通, 用户可利用高速通道控制自助完成专线接入工作。但如果涉及跨境的网络打通, 由于各国政策不同, 流程可能不一样, 用户可咨询阿里云以了解详细情况。

2. VPN 接入

阿里云和用户自建数据中心或其他云计算平台之间利用 VPN 打通网络也是一个可选方案。VPN 相对于高速通道专线接入有投入小、实施简单的优点。但其服务质量受公共网络影响较大, 对网络的稳定性和速率要求很高的业务场景不适合使用这种方案。

通常在遇到以下情形时, 建议采用 VPN 混合云接入方案:

- ❑ 远程运维管理需求。
- ❑ 云上、线下的测试、验证环境 (例如接入专线前)。
- ❑ 当有多点接入需求, 且对网络的延迟等情况要求不高的情况。
- ❑ 专线的备用应急手段。

图 3-1 所示就是一个典型的在线旅行代理商多点接入场景下的 VPN 混合云接入的架构。在线旅游代理商核心后台系统部署在阿里云 VPC 网络环境, 通过 VPN 方式和代理商线下门店进行网络互通, 保证了门店到后台核心系统之间通信线路的安全性。通过外网提供 2C 业务以及对接阿里旅行、艺龙等第三方在线旅游平台。

用户可直接从云市场上购买云上 VPN 的产品, 也可以自己在 ECS 上安装配置。只要能满足 VPN 站点到站点的需求即可。推荐用户使用阿里云云市场上的 FlexGW IPsec VPN 服务器企业版。

^① 需要阿里云技术人员配合用户进行调试会产生一次性的物理专线接入费用。

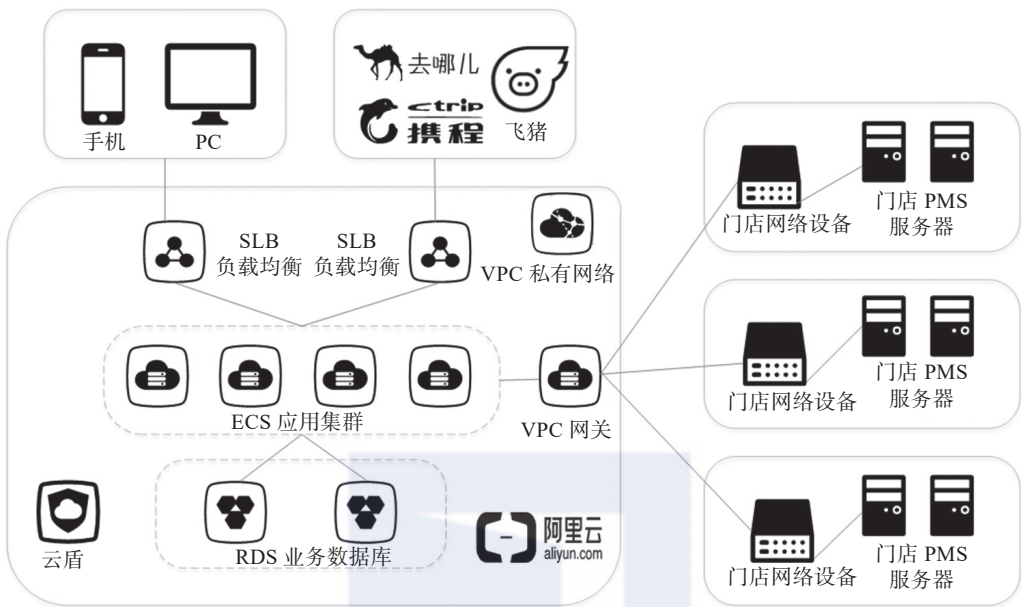


图 3-1 典型混合云架构

3.2 云上运维管理架构设计

为保证云上运维管理操作的安全性和便捷性，客户需要根据业务对云上网络环境进行规划和隔离并对内部用户访问进行管理和权限规划。下面将会从这两方面介绍如何利用阿里云产品实现云上运维管理架构设计。

3.2.1 基于 RAM 实现账号权限管理方案

本节将介绍基于阿里云 RAM 账号体系在内部用户访问资源时如何进行管理和权限规划。

1. RAM 简介

RAM（Resource Access Management）是阿里云为客户提供的用户身份管理与访问控制服务。使用 RAM，可以创建、管理用户账号（比如员工、系统或应用程序），并可以控制这些用户账号对相关资源的操作权限。当企业中存在多用户协同操作资源时，使用 RAM 可以避免用户之间共享云账号密钥，按需为用户分配最小权限，从而降低企业信息安全风险。

2. RAM 应用场景

RAM 主要的应用场景包括企业子账号管理与分权、不同企业之间的资源操作与授权管理、针对不可信客户端 App 的临时授权管理。

（1）企业子账号管理与分权

企业 A 购买了多种云资源（如 ECS 实例、RDS 实例、SLB 实例、OSS 存储桶等）。A

的员工需要操作这些云资源，但每个人的操作内容不同，比如有的负责购买，有的负责运维，还有的负责线上应用。由于每个员工的工作职责不一样，因此需要的权限也不一样。出于安全或信任的考虑，A 不希望将云账号密钥直接透露给所有员工，而是希望能给每个员工创建相应的用户账号。用户账号只能在授权的前提下操作资源，不需要对用户账号进行独立的计量 / 计费，所有开销都算在 A 的头上。当然，A 随时可以撤销用户账号的权限，也可以随时删除其创建的用户账号。

（2）不同企业之间的资源操作与授权管理

假设 A 和 B 是两个不同的企业。A 购买了多种云资源（如 ECS 实例、RDS 实例、SLB 实例、OSS 存储桶等）来开展业务。A 希望能专注于业务系统，而将云资源运维、监控、管理等任务委托或授权给企业 B。当然，企业 B 可以进一步将代运维任务分配给其公司员工完成。利用 RAM，企业 B 可以精细控制其员工对 A 的云资源操作权限。如果 A 和 B 的这种代运维合同终止，A 随时可以撤销对 B 的授权。

（3）针对不可信客户端 APP 的临时授权管理

假设企业 A 开发了一款移动 APP，并购买了 OSS 服务。该移动 APP 需要上传数据到 OSS（或从 OSS 下载数据），A 不希望 APP 通过应用服务器来进行数据中转，而希望 APP 能直连 OSS 上传 / 下载数据。由于移动 APP 运行在用户自己的终端设备上，这些设备并不受 A 的控制。出于安全考虑，A 不能将访问密钥保存到移动 APP 中。A 希望将安全风险控制到最小，比如，每个移动 APP 直连 OSS 时都必须使用最小权限的访问令牌，而且访问时效也要很短（比如 30 分钟）。这种情况下可以通过 RAM 来进行账号权限的设置。

3. RAM 账号权限设计原则

在使用 RAM 设计账号权限时，需要遵循如下原则：

（1）为根账户和 RAM 用户启用 MFA

建议为根账户绑定 MFA（多因素认证），每次使用根账户时都强制使用 MFA，以免无关人员轻易使用用户密码登录到阿里云，保证用户账号的安全性。阿里云 MFA 的详细介绍请参考 https://help.aliyun.com/Know/edge_detail/37215.html。

（2）使用群组给 RAM 用户分配权限

一般情况下，用户不必对 RAM 用户直接绑定授权策略，更方便的做法是创建与人员工作职责相关的群组（如 admins、developers、accounting 等），为每个群组绑定合适的授权策略，然后把用户加入这些群组。

（3）将用户管理、权限管理与资源管理分离

一个好的分权体系应该支持权力制衡，从而尽可能地降低安全风险。在使用 RAM 时，应该考虑创建不同的 RAM 用户，其职责分别是管理 RAM 用户、管理 RAM 权限以及各产品的资源操作管理。

（4）为用户登录配置强密码策略

如果需要允许用户更改登录密码，那么应该要求他们创建强密码并且定期更换。用户可

以通过 RAM 控制台为 RAM 用户创建密码策略，如要求最短长度、是否需要非字母字符、多长时间必须进行更换等。

(5) 定期更换用户登录密码和访问密钥

建议 RAM 用户要定期更换登录密码或访问密钥。

(6) 及时撤销用户不再需要的权限

当一个用户由于工作职责变更而不再使用某个权限时，应该及时撤销该用户的权限。

(7) 将控制台用户与 API 用户分离

不建议给一个 RAM 用户同时创建用于控制台操作的登录密码和用于 API 操作的访问密钥。通常只给员工创建登录密码，只给系统或应用程序创建访问密钥。

(8) 使用策略限制条件来增强安全性

建议给用户授权时设置策略限制条件，这样可以增强安全性。比如，授权用户 Alice 可以关停 ECS 实例，限制条件是 Alice 必须在指定时间在公司网络中执行该操作。

(9) 不要为根账户创建访问密钥

由于根账户对名下资源有完全控制权限，因此为了避免因访问密钥泄露所带来的灾难性损失，我们不建议创建根账号访问密钥并使用该密钥进行日常工作。

(10) 遵循最小授权原则

最小授权原则是安全设计的基本原则。当需要给用户授权时，应授予刚好满足他工作所需的权限，而不要过度授权。

3.2.2 云上开发、测试、生产环境管理方案

本节将介绍如何在阿里云上构架开发、测试、生产环境，从而实现不同环境之间的云资源和访问权限隔离。目前可以使用两种方案，一是使用阿里云多租户体系实现不同开发、测试、生产租户之间的资源逻辑隔离；另外一种方案是利用 VPC 基础网络产品划分多个不同的网络环境，从而实现开发、测试、生产环境资源的物理隔离。

1. 多租户逻辑隔离方案

多租户逻辑隔离方案的原理是，基于阿里云账号系统，通过申请不同的阿里云账号，利用阿里云不同账号之间资源默认无法共享和互通的特点，规划好不同账号的管理职责和权限，从而达到用户开发、测试、生产环境的隔离。多租户隔离方案是基于阿里云账号体系实现的，主要应用于需要逻辑隔离的场景。此时，客户需要维护多个阿里云账号，并进行资源管理和维护。多个阿里云账号之间完全隔离，无法实现相互调用和访问。

图 3-2 给出了某政府部门开发、测试、生产环境使用阿里云账号实现资源逻辑隔离的示意图。

2. 多 VPC 隔离方案

多 VPC 隔离方案是采用阿里云网络虚拟化产品 VPC，从而在同一个企业账号下，将开发、生产、测试资源划分到不同的 VPC 网络集群中，实现用户环境的隔离。多 VPC 隔离方

案是基于网络层实现客户不同业务环境的隔离和管理，和未上云前自建 IDC 下开发、测试、生产环境隔离方案非常相似。实施 VPC 隔离方案后，VPC 之间默认是无法通信的，但可以通过高速通道实现 VPC 打通，也可以使用阿里云网络安全组产品或云市场上的防火墙镜像，对不同网络环境之间的访问进行安全策略设置。

图 3-3 给出了某企业基于阿里云 VPC 网络环境实现开发、测试、生产环境的网络和业务隔离。

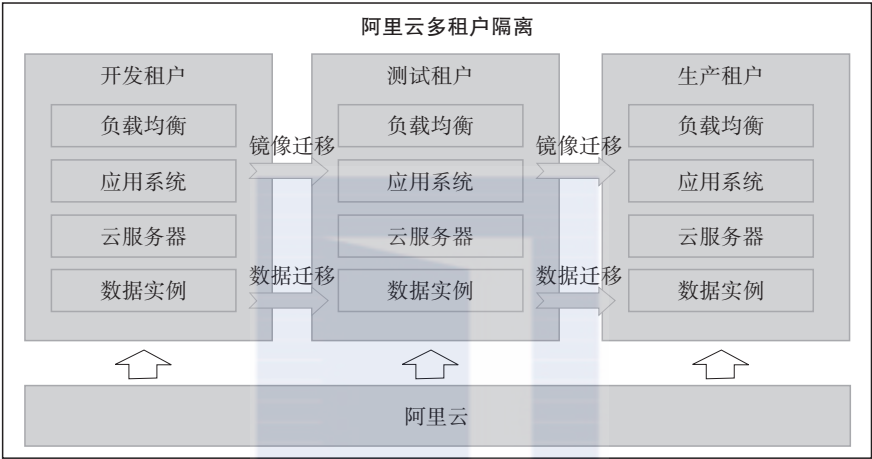


图 3-2 多租户逻辑隔离架构

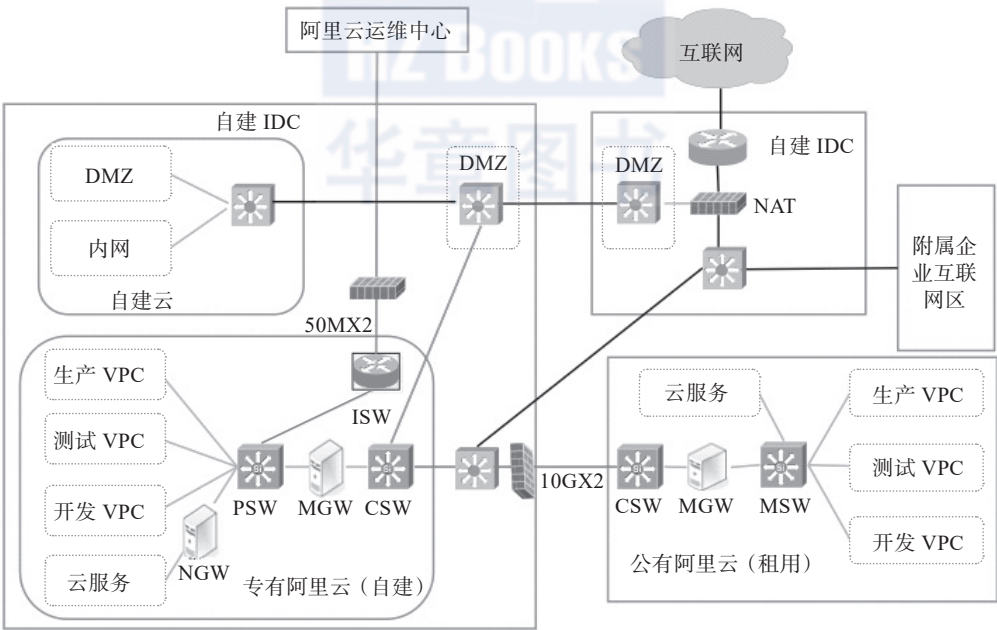


图 3-3 多 VPC 架构

3.3 应用架构设计

上面已经介绍了用户业务上云时如何进行网络设计、运维管理环境规划，本章将重点介绍如何基于阿里云产品和服务设计应用系统架构。

3.3.1 负载均衡

阿里云负载均衡（Server Load Balancer，SLB）是将访问流量根据转发策略分发到后端多台 ECS 的流量分发控制服务。用户可以通过负载均衡的流量分发扩展应用系统对外的服务能力，通过消除单点故障提升应用系统的可用性。

阿里云负载均衡主要功能：

- ❑ 负载均衡服务通过设置虚拟服务地址（IP），将多台云服务器 ECS 实例虚拟成一个高性能、高可用的应用服务池。根据应用指定的方式，将来自客户端的网络请求分发到云服务器池中。
- ❑ 负载均衡服务会检查云服务器池中 ECS 的健康状态，自动隔离异常状态的 ECS，从而解决单台 ECS 的单点问题，同时提高应用的整体服务能力。
- ❑ 在标准的负载均衡功能之外，负载均衡服务还具备 TCP 与 HTTP 抗 DDoS 攻击的特性，增强应用服务器的防护能力。
- ❑ 负载均衡服务是 ECS 面向多机方案的一个配套服务，需要同 ECS 结合使用。

在阿里云平台之上，用户应用可以使用 SLB 软负载均衡产品实现 Web 服务器和应用服务器的自动负载均衡和故障检测。阿里云 SLB 负载均衡产品可支持四层 TCP 协议和七层 HTTP/HTTPS 协议。

3.3.2 可扩展性

前面介绍了阿里云上应用系统接入层的设计，本节将介绍应用系统的可扩展性设计。应用系统可扩展性设计主要包含应用程序层的可扩展性设计和数据存储层的可扩展性设计。

1. 应用程序可扩展性设计

采用纵向扩展时，无须担心如何处理用户请求，这些请求都将汇集到单个服务器的网络接口卡上，并且得到处理，对它们的恢复也是通过网卡发送的。在不同处理器之间分配负载的具体任务，就交给操作系统调度程序来完成。如果应用程序是多进程的，可以给每个处理器分配一个子进程，然后同时处理多个请求。

但在横向扩展时，新问题出现了。我们有多处理器，却没有操作系统分配请求。我们将在同一 IP 地址收到多个请求，而且希望在多台机器上服务这些请求。

最简单解决方案是引入应用层的负载均衡方案。一种方式是在 DNS 中为应用程序的域名创建一条以上的记录。通过 DNS 轮询随机将业务请求发送给实际提供服务的主机进行处理，以扩充应用服务器的处理能力。但此方案存在一个问题，即 DNS 服务无法自动发现后

端出现异常的应用服务器，并自动屏蔽故障应用服务器对业务的影响。另外一种方式是使用阿里云 SLB 产品，作为统一接入层对外提供服务，SLB 可自动转发业务流量到不同的后端服务器进行处理，并自动检测后端应用服务器是否正常，主动发现和处理异常应用服务器。

2. 数据存储层可扩展性设计

阿里云上数据存储层的常用产品有 RDS 关系数据库服务、OSS 对象存储服务和 Redis 数据缓存服务。OSS 对象存储服务可动态扩展存储容量和访问量，无须用户在业务层考虑容量的可扩展性设计。下面将重点介绍 RDS 关系数据库服务和 Redis 数据缓存服务应如何实现可扩展性设计。

Redis 数据缓存服务可支持主从版和集群版两种实例，如果用户业务在存储量或访问量急剧增加，已超过之前设计容量时，用户可以考虑通过扩展 Redis 实例规格或从主从版实例升级为集群版实例，来实现 Redis 数据缓存在存储容量和访问量方面的能力提升。用户使用阿里云 Redis 服务时，如果业务遇到实例访问量 QPS 的 CPU 瓶颈，需要通过扩充 Redis 实例节点数目来解决，如考虑从主从版 Redis 实例升级为集群版 Redis 实例，或从低规格的集群版 Redis 实例升级高规格集群版 Redis 实例。

RDS 关系数据库服务的扩展能力，主要从以下几个维度考虑：

- ❑ 升级 RDS 实例规格：可在一定程度上扩展 RDS 读写请求的处理能力和存储容量。
- ❑ 引入 RDS 只读实例：使用 DRDS 实现自动读写分离，可扩展 RDS 数据库的读流量，适用于读多写少的业务场景。
- ❑ 引入 Redis 进行热点数据缓存：这同样适用于读多写少和高性能、低延时的场景，需要业务代码能维护好 Redis 缓存和 RDS 关系数据库的数据一致性。
- ❑ 使用 DRDS 分库分表功能实现 RDS 数据库存储容量和读写访问量的可扩展性设计：这适用于只读实例、缓存、实例规格升级无法解决的场景。由于 DRDS 引入了分布式存储技术，需要对业务进行合理规划和设计尽量避免分布式问题。

3.3.3 微服务

1. 微服务与传统的集中化开发方式的差异

在传统的开发模式下，所有的功能打包在一个 WAR 包里，基本没有外部依赖，部署在一个 J2EE 容器（Tomcat、JBoss、WebLogic）里，包含 DO/DAO、Service、UI 等所有逻辑。所有的逻辑组合成一个庞大的运维单元，独立进行开发、升级、测试、部署等。

而微服务架构主张互联网应用由一系列细粒度的职责单一的服务组成，每个服务运行在各自独立的进程中，服务之间通过轻量机制（如 HTTP/JSON）进行通信。这些服务建立在业务领域之上，可通过全自动方式独立部署。这些服务基本没有中央式的管理，服务可以用不同的编程语言编写，也可以使用不同的数据存储技术。微服务的目的是有效地对拆分的应用进行服务治理，简化部署，实现敏捷开发和部署。

然而，微服务不是免费的午餐。微服务系统复杂度高，运维开销大，需要完善的服务框架、交付、监控、安全隔离和规模扩展等必要的基础设施的支撑，对于 DevOps 技能要求高。对于初创和中小规模公司来说，借助于阿里云的基础设施来实现微服务的治理框架和持续交付，不失为最佳选择。

2. 服务治理框架的选择

(1) 使用 Spring Cloud 构造微服务

Spring Cloud 是 Springframework 里的一个项目，提供了开发分布式系统时比较常见的一些模式能力。其中，配置管理（Config Server）、服务发现（Eureka）、服务熔断（Hystrix）、智能路由（Zuul）等是基于 Netflix OSS 的一个封装，利用 Java 注解（annotation）声明，可以在 Spring Boot 应用里便捷地使用 Netflix 的开源产品构建生产级可用的微服务应用。而且，可以通过 Docker 镜像的方式快速地在阿里云容器服务上部署。

(2) 使用 EDAS 来构造微服务

EDAS 产品充分利用阿里云的资源管理和服务体系，引入阿里巴巴中间件整套成熟的分布式产品，全面兼容 Apache Tomcat 的 Java 容器，提供高性能的分布式服务框架以及秒级推送的分布式配置管理服务。此外，EDAS 还创新性地提供了分布式系统链路追踪、容量规划、数据化运营等多款组件，可以帮助企业级客户轻松构建大型分布式应用服务系统。

在复杂的云环境下，应用发布与管理变得十分复杂。本地开发完成的应用需要逐个部署到服务器，然后登录每一台服务器终端进行应用的发布和部署；后续还会有应用的重启、扩容等工作需要完成。服务器的不断增加对于运维人员是一个极大的挑战。针对这个场景，EDAS 提供了一个可视化的应用发布与管理平台，无论集群规模多大，都可以在 Web 控制台上轻松地进行应用生命周期管理。

当从集中式应用转变成分布式系统的时候，系统之间的可靠调用一直都是分布式架构的难题，比如需要确定网络通信、序列化协议设计等很多技术细节。EDAS 提供了一个高性能的 RPC 框架，能够构建高可用的分布式系统，系统地解决各个应用之间的分布式服务发现、服务路由、服务调用以及服务安全等细节。

应用开发完毕部署到生产环境之后，通常需要对应用运行时状态进行监控，比如监控 CPU 使用率、机器负载、内存使用率和网路流量等。但此类基础监控通常满足不了业务需求，比如系统运行变慢却无法定位瓶颈所在，或者页面打开出错但是无法排查具体调用错误等。对此，EDAS 提供了一系列系统数据化运营组件，针对分布式系统的每一个组件和每一个服务进行精细化的监控和跟踪，建立数字化分析系统，从而精准地找到系统瓶颈。

3. 如何通过各种构建物实现持续交付方案

(1) 使用阿里云持续交付平台 CRP 系统来进行代码持续交付

阿里云持续交付平台 CRP 支持以鼠标在白屏上拖拽节点的方式定义项目发布 workflow，每个节点可以加入多个任务，用以完成自动化更新代码、编译、运行单元测试、自动化发布

到 ECS 机器上的工作。

如果需要基于代码库，执行代码扫描（安全检查）→自动化编译→测试→自动化部署到服务器的工作时，可以在持续交付平台 CRP 上定制一条持续发布线。持续发布线定制完成后，当代码更新后，CRP 会监听到分支更新了代码，自动创建 1 条新的发布线开始运行，自动进行编译、测试、部署等工作。并且出现问题时，可以发邮件通知项目成员。

不同的技术栈有不同的构建物类型，同时也有相关的工具来创建和安装这些构建物，例如，Ruby 中有 gem、Java 中有 WAR 包、Python 中有 egg。我们可以通过 CRP 等工具部署并且启动这些构建物，但是需要安装和配置一些基础环境软件。类似于 Puppet、Chef、ansible 等自动化配置管理工具，可以很好地解决这个问题。通过这些工具可对不同构建物的底层部署机制进行屏蔽。

（2）ECS 定制化镜像

使用类似 Puppet、Chef 及 Ansible 这些自动化配置管理工具的一个问题是，需要花费大量时间在机器上运行这些脚本。假设服务器在 ECS 上，使用的是标准的 CentOS 镜像，为了运行 Java 应用程序，需要做的第一件事情就是安装 Oracle JVM，接下来需要安装 Tomcat 容器。随着集群规模的扩展，同样的工具被一遍一遍地重复安装，对技术人员而言也是一种煎熬。

减少扩容成本的一种方法就是创建一个虚拟机镜像，镜像云服务器 ECS 实例运行环境的模板，一般包括操作系统和预装的软件。可以使用镜像创建新的 ECS 实例和更换 ECS 实例的系统盘。使用这种功能之后，事情就变得简单了。现在可以把公共的工具安装在镜像上，然后在部署软件时，只需要根据镜像创建一个实例，然后在其上安装最新的服务版本即可。也就是说，只需要构建一次镜像，然后根据这些镜像启动 ECS，就不需要再花费时间来安装相应的依赖，因为它们已经在镜像中安装好了，这样可以节省很多时间。如果核心依赖没有改变，那么新版本的服务就可以继续使用相同的基础镜像。

当然，既然已经做到了使用包含依赖的虚拟机镜像来加速交付，那么还可以更进一步，把服务本身也包含在镜像中，直接把镜像作为构建物进行交付，通过阿里云 ECS 的创建自定义镜像功能，利用自定义镜像安装 ECS，当虚拟机启动镜像时，服务就已经就绪了。

通过镜像交付可以屏蔽操作系统和软件的差异性，但是如果是一次交付流程中，从开发测试、预生产、生产环境，每个环节都要制作自定义镜像、重新安装操作系统，对于迭代速度非常频繁的应用，那么虚拟机镜像交付这种略显笨重方式显然是难以接受的。

如果想既交付服务代码又要保证应用环境和操作系统的差异性，同时希望以敏捷、轻量级的方式交付，那么阿里云容器服务能很好地满足这个需求。利用容器技术把应用及其依赖做成一个标准的镜像，从开发到测试到生产环境都用同样的容器镜像，极大弥补了应用开发过程中开发和运维之间的交付鸿沟。

（3）容器服务定制化 docker 镜像

阿里云容器服务（Container Service）提供了高性能可伸缩的容器应用管理服务，支持在一组云服务器上通过 Docker 容器来进行应用生命周期管理。容器服务极大简化了用户对

容器管理集群的搭建工作，无缝整合了阿里云虚拟化、存储、网络和安全能力，从而实现云端优化的运行环境。容器服务提供了多种应用发布方式和流水线般的持续交付能力，原生支持微服务架构，可帮助用户无缝上云并进行跨云管理。

传统的开发过程开发者的代码里有逻辑、应用以及代码依赖包，通过使用阿里云容器服务，代码中会加入 Dockerfile、Docker Compose，用来制作集装箱和搬运集装箱。代码提交成功后，代码服务器会通知持续集成服务（比如 Jenkins），持续集成服务会拉取代码进行代码打包，打包后进行单元测试。如果单元测试没有通过，就会马上向开发者反馈。如果通过测试，则会根据代码中的 Dockerfile 制作镜像，并把镜像推送到阿里云容器仓库服务，并用于应用交付。在 Docker 容器镜像中可以支持不同类型的应用，无论是 PHP 应用还是 Java 应用，也就是说，不同种类的应用可以用相同的命令或 API 进行管理，规避了不同软件之间的差别，标准化了软件的构建、交付和运维方式。

3.3.4 异步化

在高并发量下，微服务中如果采用同步调用方式，在服务单元处理时间变长的情况下，会导致服务线程消耗尽，遇到无法再创建线程处理服务请求的情况。对于这种情况的优化，除了在程序上不断调优（数据库调优、算法调优、缓存等），可以考虑在架构上做些调整，先返回结果给客户端，让用户可以继续使用客户端的其他操作，再把服务端的复杂逻辑处理模块做异步化处理。

比如，一个下单系统的展现端的逻辑强依赖后端 10 个服务，包括减库存、生成快照、运查询费、优惠查询等，其中 9 个都执行成功了，最后一个却执行失败，那么是不是前面 9 个都要回滚掉？如果这样做，成本还是非常高的。所以在拆分大的流程为多个小的本地事务的前提下，对于非实时、非强一致性的关联业务，在本地事务执行成功后，我们选择通过消息机制将关联事务异步化执行，从而保障整个业务流程的高效运转。

在阿里云上，可以使用阿里云提供的消息队列服务（Message Queue，MQ），帮助业务开发人员完成业务流程异步化。MQ 消息队列服务是企业级互联网架构的核心产品，基于高可用分布式集群技术，搭建了包括发布订阅、消息轨迹、资源统计、定时（延时）、监控报警等一套完整的消息云服务。MQ 可以帮助用户实现分布式计算场景中服务的异步化调用，帮助服务解耦并进行异步化改造。MQ 目前提供 TCP、HTTP、MQTT 三种协议层面的接入方式，支持 Java、C++ 以及 .NET 等编程语言，方便用不同编程语言开发的应用能快速接入 MQ 消息云服务。用户可以将应用部署在阿里云 ECS、企业自建云，或者嵌入到移动端、物联网设备中，与 MQ 建立连接并进行消息收发，同时，本地开发者也可以通过公网接入 MQ 服务进行消息收发。

3.3.5 高可用

建设应用系统时，特别是建设一些核心和关键的业务系统时，我们需要考虑和评估一些

极端异常的场景或故障发生时对整个应用系统和业务的影响。例如，单个机房出现网络或电力故障导致机房无法对外提供服务、某个城市地震、台风等自然灾害等情况。

针对这些极端场景或问题，阿里云提供“同城双机房”和“两地三中心”的灾备高可用解决方案，通过在同城或者异地建立两套或多套功能相同的应用系统，并集成健康监测和灾备切换功能，当一处系统因意外（如火灾、地震等）停止工作时，整个应用系统可切换到另一处备用系统，继续对外提供服务。

1. 同城双机房高可用设计

图 3-4 给出了同城双机房架构。其中包括 SLB 网络接入层、ECS 云服务器 Web 层和应用层、RDS 数据库层以及 OSS 文件存储层。

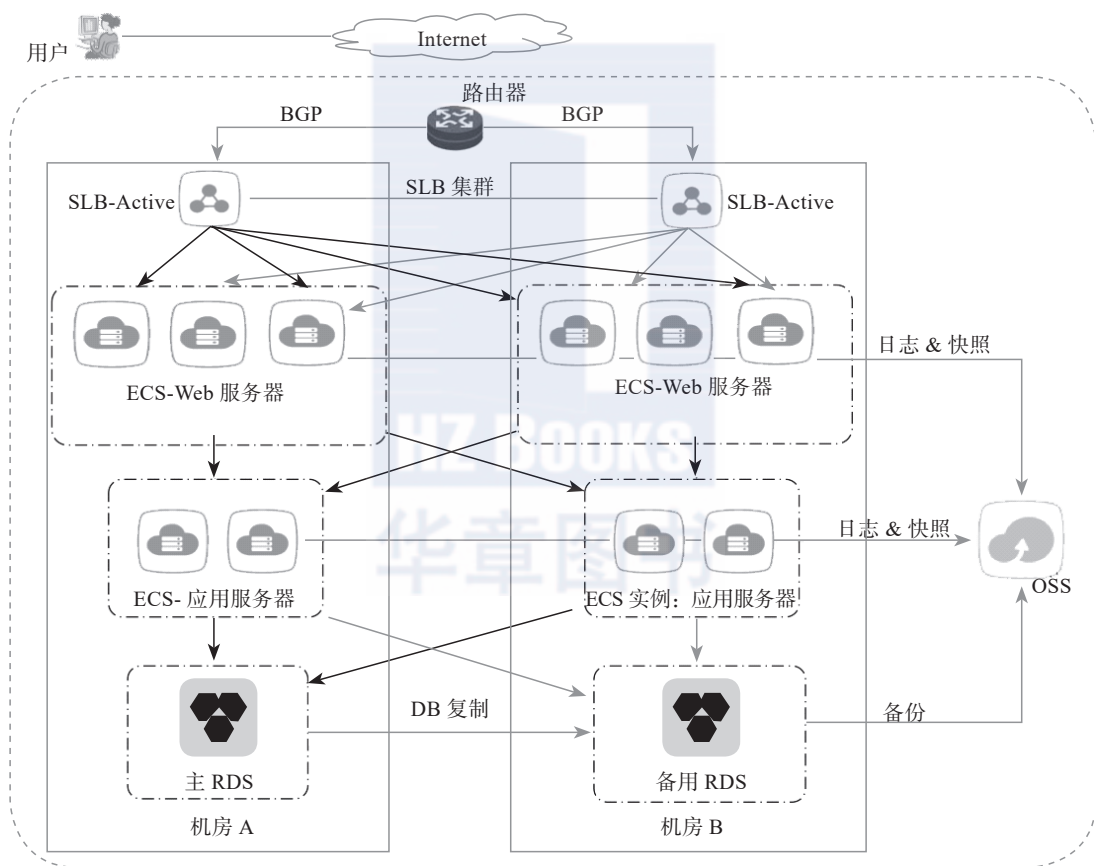


图 3-4 同城双机房高可用架构

1) SLB 网络接入层：用户通过 Internet 访问 BGP SLB，SLB 集群根据用户来源 IP 段，将流量分配到不同机房的 SLB 节点上，实现机房间 SLB 双活。

2) ECS 云服务器 Web 层和应用层：在不同机房分配 ECS 资源，在不同机房完成对等

的 Web 和应用服务器部署，所有的 Web 服务器分别挂载到对应机房的 SLB 上。

3) RDS 数据库层：RDS 采用主备库模式双机房部署，数据实现近实时同步。Web & APP 服务器正常情况下只访问 RDS 的 Active 的主库。

4) OSS 文件存储层：非结构化数据将存入 OSS，如 ECS 的系统快照和 RDS 的数据库备份等静态数据会统一上传到 OSS 存储集群上。

2. 两地三中心高可用方案

图 3-5 给出了两地三中心架构。其中，最上层是 DNS SLB 网络接入层，用户通过外网 DNS 访问在线集群的 SLB，DNS 引流到活动的主站点。中间是由 ECS 集群组成的 Web 层和应用层，该层在不同地域、不同机房分配 ECS 资源，并完成 Web 和应用服务器部署。不同地域的 Web 和应用服务器分别挂载到相同地域的 SLB 实例之上。最下面是 RDS 数据库层，同一地域 RDS 采用主备库模式双机房部署，数据同步采用 RDS 主备复制方案，不同地域的 RDS 之间的数据同步使用阿里云数据传输产品实现秒级复制。Web 和 APP 服务器正常情况下只访问活跃的 RDS 主库。

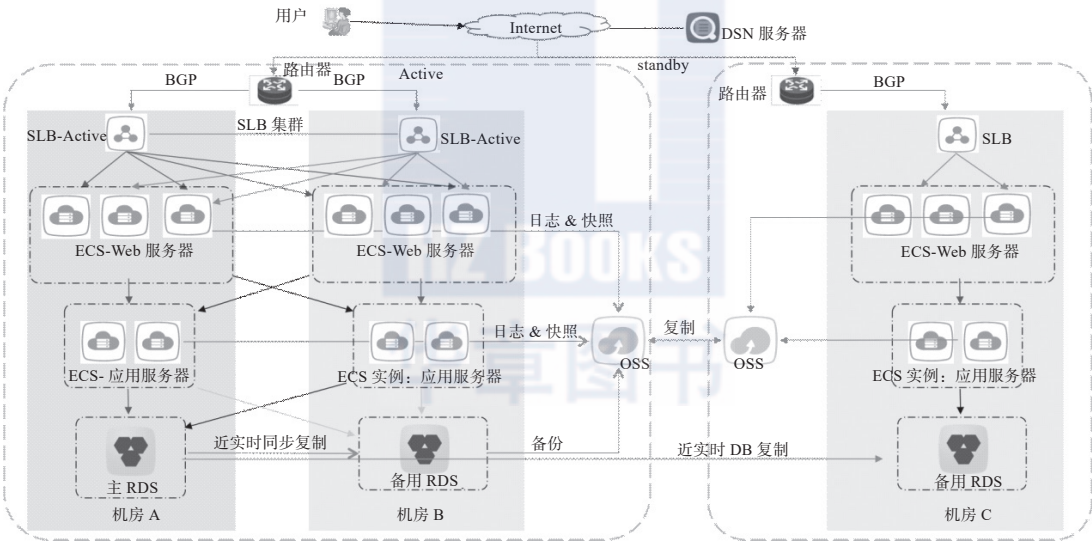


图 3-5 两地三中心容灾架构

OSS 文件存储层负责快照文件、备份文件等非结构化数据存储，如 ECS 操作系统的快照、RDS 数据库的备份文件等静态数据，存储到 OSS 存储服务上。不同地域 OSS 数据容灾采用 OSS 数据迁移服务完成。

当机房发生故障时，会优先切换到本地域的不同机房。当因主干网络或其他原因导致整个地域业务受影响时，会将业务切换到不同地域的机房。

3.3.6 云数据库架构最佳实践

前面几节分别从应用负载均衡、可扩展性、微服务架构、异步化、高可用等技术层面做了介绍，本节将结合典型应用场景介绍整体技术架构的设计与实现。

1. 小型 OLTP 系统

小型 OLTP 系统是指数据量小（小于 500GB）、单表数据量级别（千万级）、访问量小、操作简单的系统，这类系统主要用于在线处理业务、简单实时数据统计和定期的数据报表（统计的数据量小（10w 级）、确定性强的查询）。

针对这类系统，可采用单机 RDS+DRDS 方案。当遇到数据库读压力过大，可以通过增加 RDS 只读实例，采用 DRDS 将部分读流量切到只读实例来解决问题。

这类系统的典型应用场景包括财务、OA 办公类系统。图 3-6 给出了小型 OLTP 的架构。

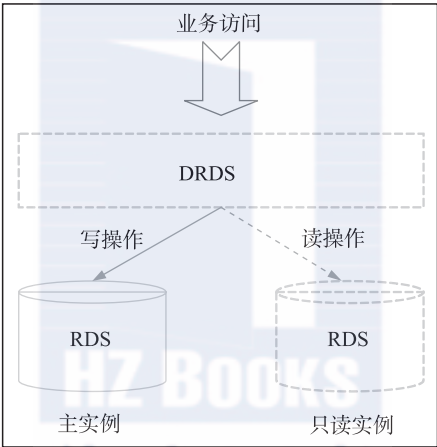


图 3-6 小型 OLTP 系统架构

2. 中型 OLTP 系统

中型 OLTP 系统是指数据量中等规模（大于 500GB 但小于 2TB）、单表数据量级别（千万以上到 1 ~ 2 亿内）、写访问量较小但读访问量较大、较复杂的系统，主要用于在线处理业务、实时数据统计和定期的数据报表（统计的数据量小（50w 内）、确定性较强的查询）。

对于这类系统，通常采用 DRDS+ 单节点 RDS+ 只读 RDS 方案，DRDS 用于对大表进行分表和自动读写分离。

该类系统的典型应用场景包括市级政务平台、互联网金融平台等。图 3-7 给出了中型 OLTP 系统架构。

3. 大型 OLTP 系统

大型 OLTP 系统是指数据量大（大于 2TB）、单表数据量级别（亿级以上）、写访问量超过单机 RDS 容量、较复杂的系统，主要用于在线处理业务、实时数据统计和定期的数据报

表（统计的数据量小（100w 内）、确定性较强的查询）。

大型 OLTP 系统多采用 DRDS+ 多个 RDS 实例的方案，DRDS 进行分库分表和自动读写分离。

大型 OLTP 系统的典型应用场景包括部委级业务平台、企业总部级应用系统。图 3-8 给出了大型 OLTP 系统架构。

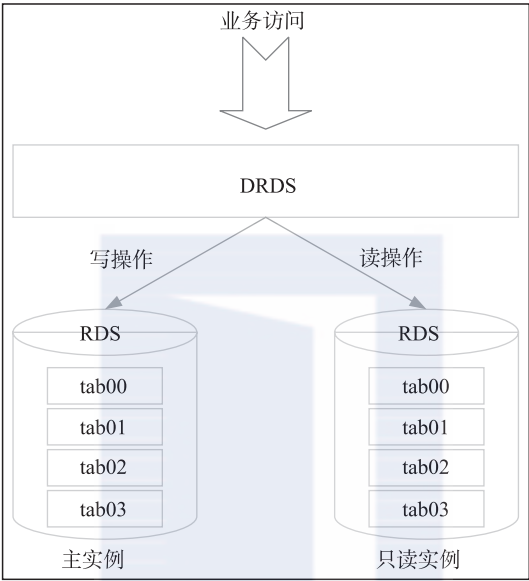


图 3-7 中型 OLTP 系统架构

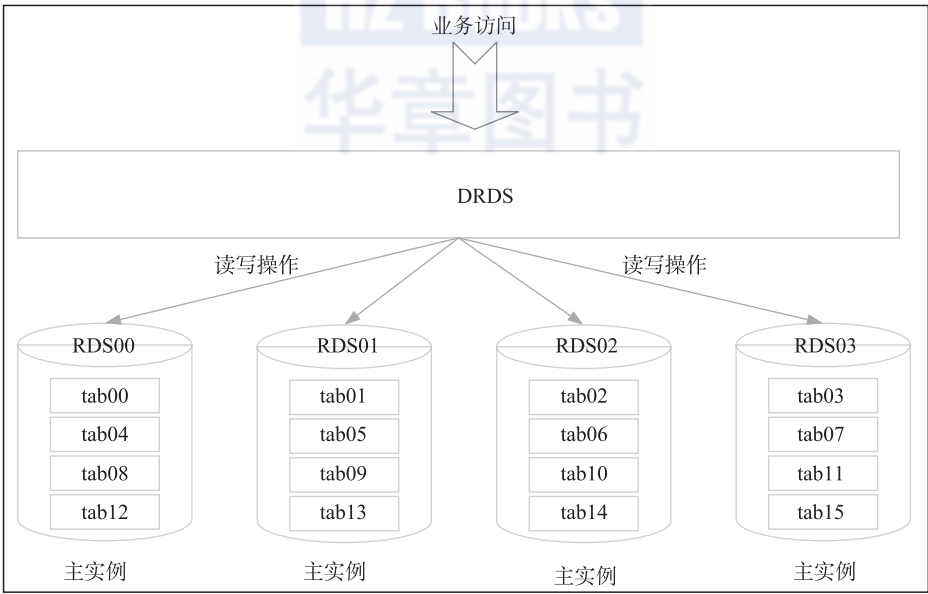


图 3-8 大型 OLTP 系统架构

4. 在线、离线一体化架构

综合型业务系统中的数据量大（超过 RDS 实例存储容量 2TB）、单表数据量极大（达到亿级、十亿级甚至百亿级），需要处理各种复杂的实时、定时大数据报表需求。

综合型业务系统通常包括在线交易、准实时大数据分析和离线大数据分析三个部分。其中在线交易部分采用 DRDS+ 多个 RDS 实例的方案，准实时大数据分析部分采用 AnalyticDB，离线大数据分析部分采用 MaxCompute。综合型业务系统的典型应用场景包括各类国家级综合业务平台。图 3-9 给出了在线、离线一体化架构。

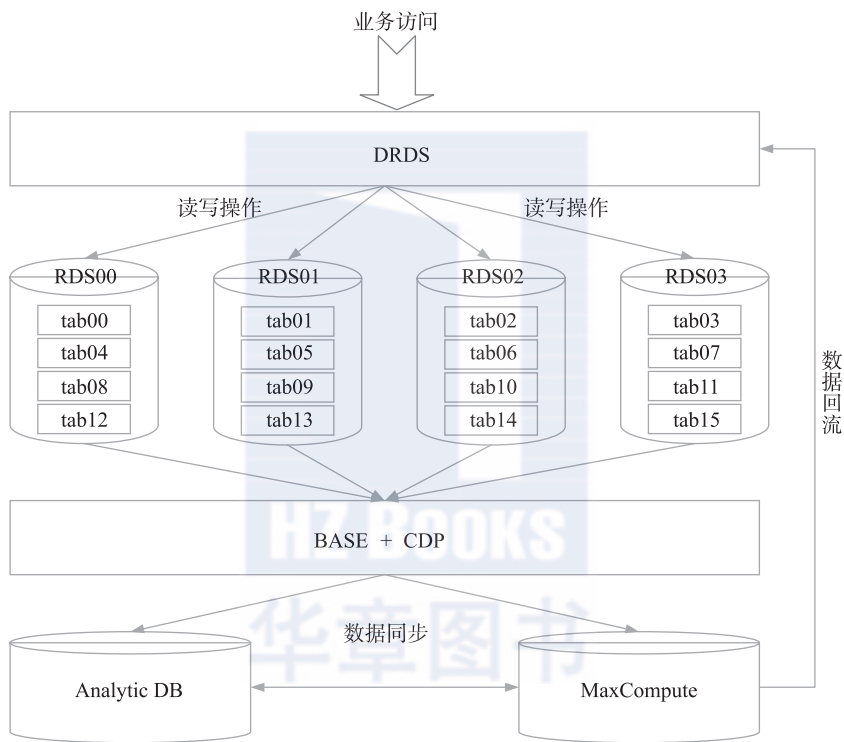


图 3-9 在线、离线一体化架构

3.4 应用改造实践

在完成阿里云上应用系统架构设计后，开始进入实施阶段，对应用系统进行架构和代码级的编码改造实施工作。

3.4.1 应用架构改造

在应用架构改造中，主要涉及负载均衡改造、Web 和应用层改造、服务化改造几方面问题。

1. 负载均衡改造

原有系统 Web、应用服务器采用硬件负载均衡设备 F5 或开源 LVS 实现 Web、应用服务器负载均衡。迁云时需要改造为云上 SLB 负载均衡服务。云上 SLB 服务可以支持 TCP、HTTP、HTTPS 等三种协议实现流量负载均衡，同时支持自动对 Web、应用服务器进行健康检查，自动屏蔽异常状态的服务器，在服务器恢复正常后自动解除屏蔽重新提供服务。如图 3-10 所示，整个业务架构中将硬件负载均衡设备 F5 或自建 LVS 集群改造为 SLB 负载均衡服务。

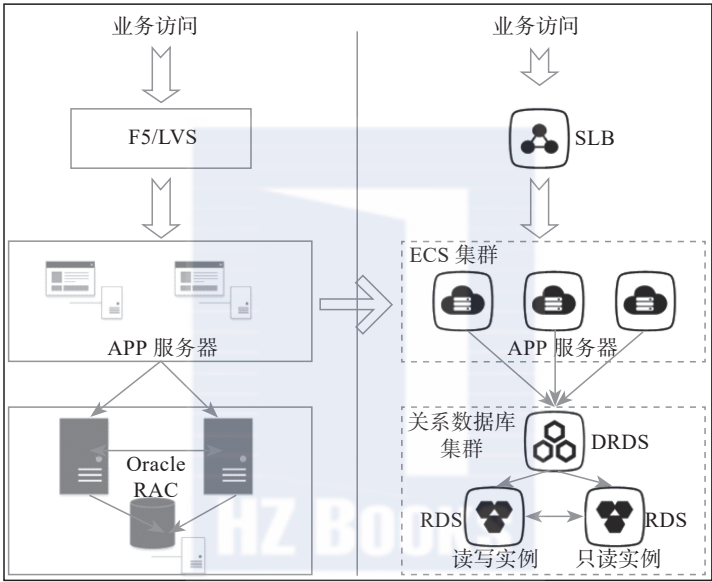


图 3-10 负载均衡架构改造

2. Web 和应用层改造

如果原系统 Web、应用部署在小型机、PC Server、商业或开源虚拟化服务器上，则可直接部署在云上弹性计算服务 ECS 上。目前 ECS 可支持多种 Linux 和 Windows 类型的操作系统。为保证 Web、应用服务的高可用性，建议同一类服务至少部署在两台 ECS 服务器上，使用 SLB 实现负载均衡和服务容错。图 3-11 给出了 Web 和应用层架构改造的示意图。

3. 服务化改造

对应用系统进行服务化改造，可使用消息中间件来实现业务系统的解耦，通过对数据库按不同业务进行拆分提升数据库的性能。例如，淘宝的系统架构充分使用了服务化设计思想和消息中间件产品实现业务系统的服务化设计 and 解耦。图 3-12 给出了业务服务化改造的过程。

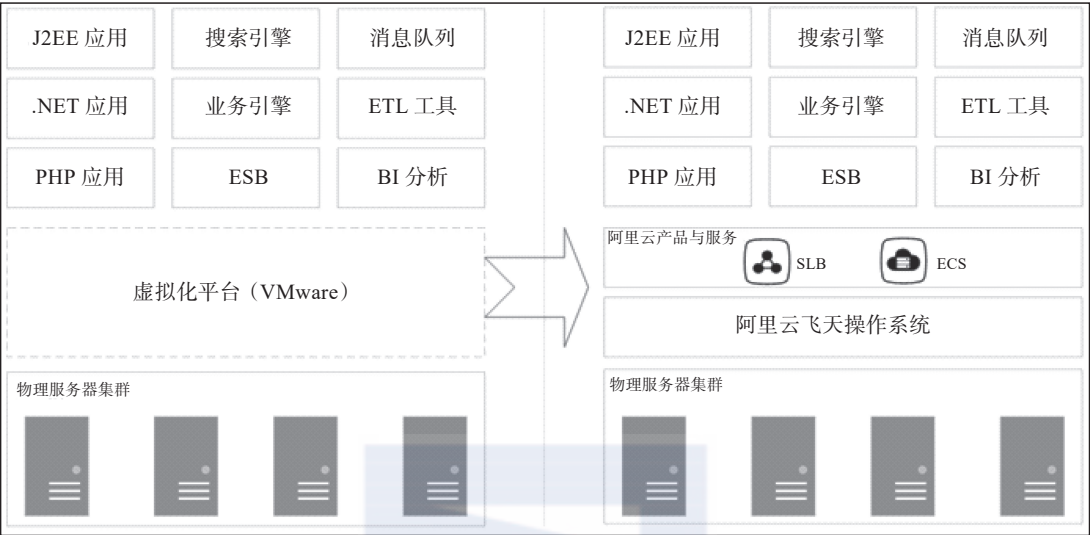


图 3-11 Web 和应用层架构改造

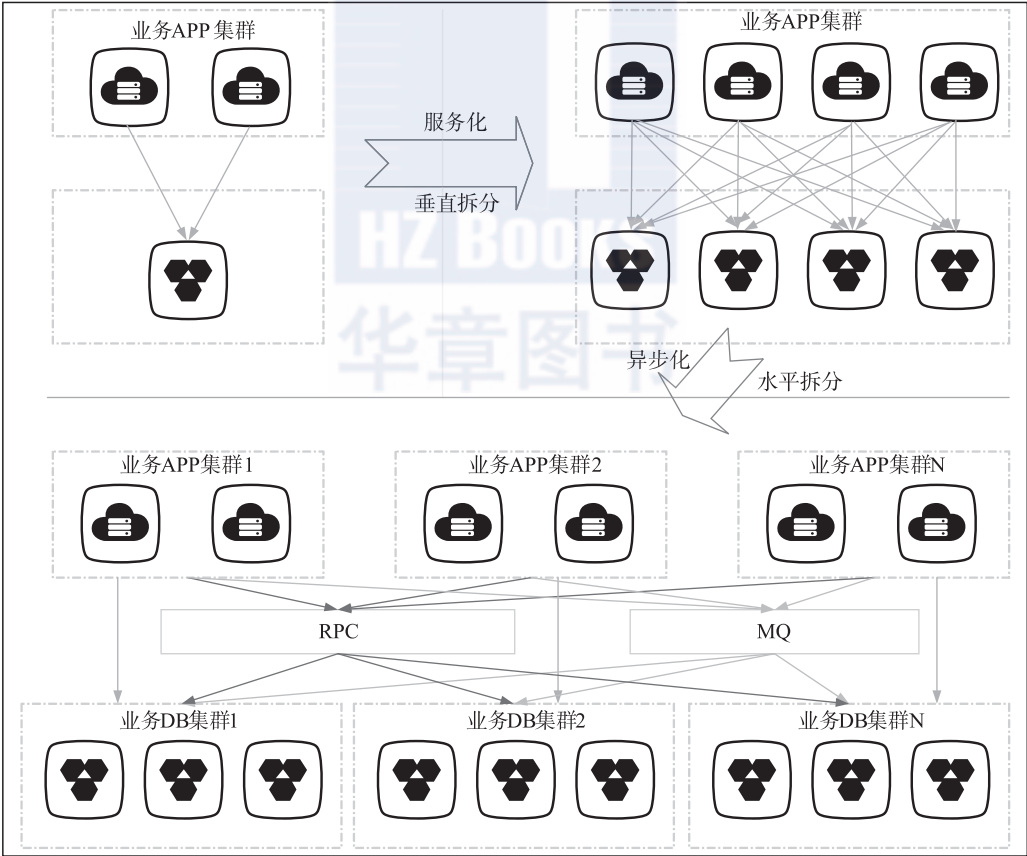


图 3-12 业务服务化异步化改造过程

在图 3-12 的架构图中，消息中间件主要的应用场景包括：

- ❑ 分布式事务：为面向服务架构实现最终一致性的分布式事务提供支持，保证全局数据的最终一致性。
- ❑ 延时队列：把消息中间件当做可靠的延迟队列。
- ❑ 广播通知：把消息中间件作为可靠的集群内广播通知，用于通知 cache 失效等事件。

3.4.2 数据库改造

数据库改造分为 OLTP 类关系数据库架构改造和 OLAP 类关系数据库架构改造，下面分别介绍这两类情况下的改造方案。

1. OLTP 类关系数据库架构改造

对于访问量比较小的基于关系数据库的系统，系统架构如图 3-13 所示。

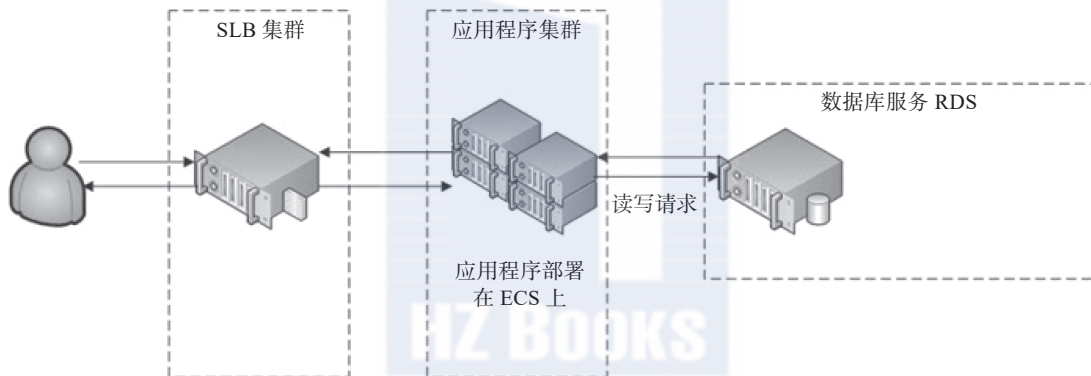


图 3-13 关系数据库架构

随着业务压力不断增大，单个 RDS 的读写已经无法满足业务访问请求的时候，尤其是读压力非常大的系统，可以考虑使用阿里云 Memcache 或 Redis 缓存系统来分摊读压力，通过缓存热点数据来提供快速访问，详细架构如图 3-14 所示。

对于读压力较大的场景也可以考虑通过增加 RDS 只读实例，将系统改造成读写分离的应用架构。特别是对于有些数据分析类的场景，读写分离也是非常有效的解决方案之一。采用读写分离架构对业务读能力进行扩展改造时，业务人员需要对业务系统进行评估，将系统中对延时不敏感的业务切换到只读实例访问。数据库读写分离架构如图 3-15 所示。

考虑到 RDS 存储容量不超过 2T 的限制，以及 RDS 单实例的性能瓶颈，当数据库的单实例存储空间无法满足或写入的 TPS 接近数据库能力上线时，数据库通常需要做垂直与水平拆分，垂直拆分是指根据不同业务类型或功能将数据库拆分到不同的 RDS 实例存储，水平拆分是指将同一类业务或数据通过一定的路由规则存储到不同的 RDS 实例中。数据库拆分的难度通常比较大，需要考虑如何做业务切分。当业务切分后，数据会分布到多个节点，

导致一系列分布式问题，如分布式事务、跨库查询、全局唯一 ID 等，所以在进行数据库架构设计或改造的过程中，我们应尽量减少拆分，降低应用实现的难度。如必须拆分，可遵从先做服务化改造以进行数据库垂直拆分，再考虑进行数据库水平拆分。

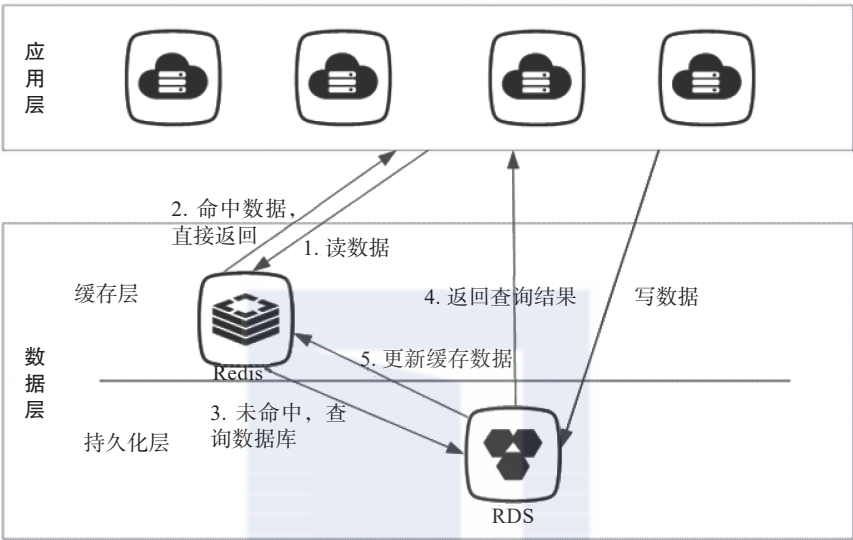


图 3-14 基于缓存优化关系数据库性能

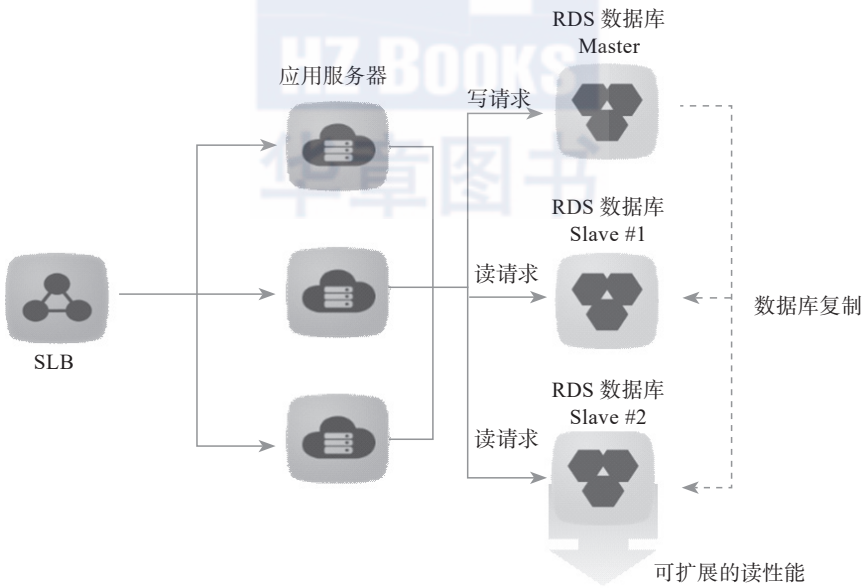


图 3-15 基于读写分离扩展关系数据库性能

阿里云平台上的分布式数据库服务 DRDS 提供数据库水平拆分和读写分离功能。它尽

量屏蔽了分布式对应用开发的影响，同时也帮助用户在一定程度上解决了数据库水平拆分导致的全局唯一 ID、跨库查询、分布式事务等问题。DRDS 数据库水平拆分和读写分离原理如图 3-16 所示，同一张表的数据通过一定路由策略和算法均衡的分布到多个 RDS 读写实例，并通过 DRDS 的读写分离功能自动帮用户将读流量路由到只读 RDS 实例。

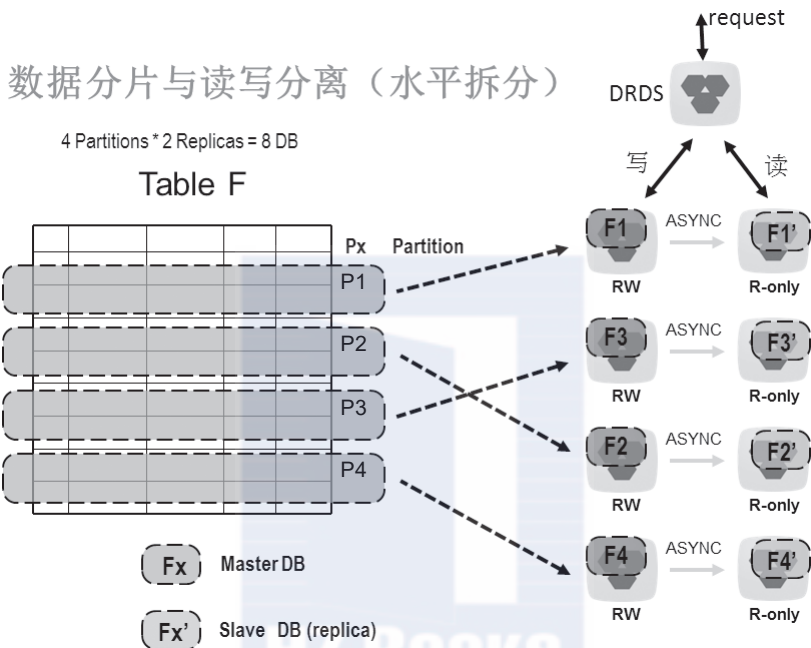


图 3-16 基于水平拆分扩展数据库性能

2. OLAP 类关系数据库的架构改造

OLAP 联机分析处理类型系统是数据仓库系统最主要的应用类型，用于支持复杂的分析操作，侧重对决策人员和高层管理人员的决策支持，可以根据分析人员的要求快速、灵活地进行大数据量的复杂查询处理，并且以一种直观而易懂的形式将查询结果提供给决策人员，以便他们准确掌握企业的经营状况，了解客户的需求，制定正确运营方案。

阿里云针对 OLAP 联机分析处理类应用的规模大小有不同的解决方案：

- ❑ 小规模分析系统：这类 OLAP 系统仅针对具体某一类业务的历史数据进行离线和实时数据分析，一般数据量在 1TB 以内，分析所使用的数据在十个维度以内，对数据分析的时效性要求较低。对于这类应用系统，一般可直接采用 RDS 关系数据库作为底层的数据存储系统，并在其上构建 OLAP 分析工具。
- ❑ 大规模实时数据分析：这类 OLAP 系统面向数据存储规模在 TB 级别以上，单表记录数达到千亿级别，业务对数据分析时效性要求较高，一般需要秒级内返回结果的系统可通过阿里云实现大规模数据分析存储系统 AnalyticDB 构架实时分析系统。

- ❑ 大规模离线数据分析：这类 OLAP 系统面向数据存储规模在 TB、PB 级别以上，单表记录数亿级以上，需要做非常复杂的计算操作，业务对数据分析时效性要求较低，一般在分钟级返回结果的系统。此类业务可基于阿里云 MaxCompute 大计算产品构建离线分析系统。

3.4.3 文件存储改造

传统行业通常采用 SAN、NAS、关系型数据库存储文件或图片等数据。但 SAN 方案成本过高且扩展性受限，NAS 无法保证高性能，关系型数据库存储文件成本高且严重影响应用系统性能，因此给存储带来一定困难。阿里云 OSS 开发存储服务（Open Storage Service）能提供海量、安全、低成本、高可靠的云存储服务，适合存放文件、图片等非结构化数据。同时，OSS 提供 Java、Python、PHP、C# 语言的 SDK。用户可以通过调用对应开发语言的 API，在任何应用、任何时间、任何地点上传和下载数据，也可以通过 Web 控制台对数据进行管理。

在 OSS 中，用户的每个文件都是一个 Object（对象），Object 包含 key（关键字）、data（数据）和 user meta（用户元组）。key 是 Object 名，user meta 是对数据的描述。存储在 OSS 上的每个 Object 必须都包含在 Bucket 中，Bucket 名在整个 OSS 中具有全局唯一性且不能修改。

从其他文件存储产品迁移至 OSS 上时，只需要使用阿里官方提供 SDK 和 API 接口替换原有接口即可。

OSS 使用的场景包括存储文件、图片、视频等非结构化数据；存储应用系统日志、数据备份；替换用户自建文件存储系统。

图 3-17 给出了 OSS 的业务架构，基于 OSS 构建面向文件对象的存储系统架构的特点如下：

- ❑ OSS 负责存储文件、照片、视频等海量非结构化数据。
- ❑ RDS 负责存储业务系统所有的结构化数据，包含文件、照片、视频等非结构化数据 OSS 访问地址或索引。
- ❑ 基于 OSS 构建独立的文件存储管理系统，应用程序可使用 SDK 或 API 方式与 OSS 存储服务对接，实现对文件、照片、视频等非结构化数据的管理。在整个文件系统的存储架构中，应用系统设计者无须考虑系统的高可用、性能和扩展能力，这些都会由底层 OSS 对象存储服务来保证。

（1）OSS 主要访问接口

OSS 为用户提供数据存储服务，用户可以通过以下 API 接口和 SDK 来管理和维护 OSS 上的数据：

- ❑ OSS Bucket 相关接口：可支持 Bucket 创建、查看、罗列、删除以及修改、获取 Bucket 的访问权限等操作。
- ❑ OSS Object 相关接口：可支持 Object 对象上传、查看、罗列、删除，并支持大文件分片上传等操作。

- ❑ OSS 访问相关：支持 If-Modified-Since 和 If-Match 等 HTTP 参数。
- ❑ OSS 支持 Object 生命周期管理，用户可自定义 OSS 上存储数据的过期时间，在到期后，OSS 会自动帮助用户完成过期数据清理和空间回收。
- ❑ OSS 支持防盗链，采用基于 HTTP header 中表头字段 referer 的防盗链方法。
- ❑ OSS 支持使用 STS 服务临时授权和 URL 签名授权访问。
- ❑ OSS 提供相应接口，方便开发者控制跨域访问的权限。

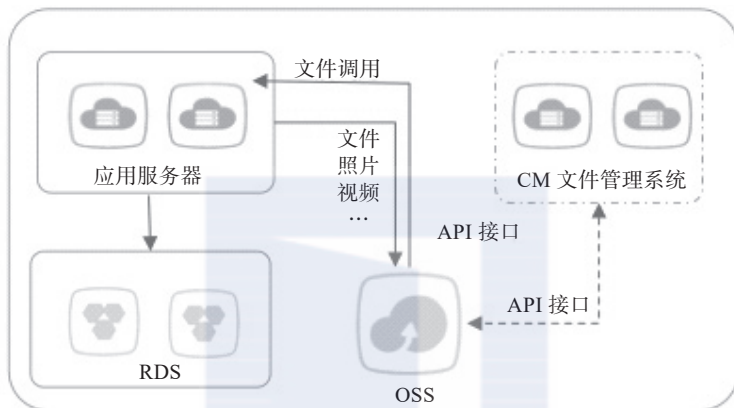


图 3-17 OSS 业务架构

(2) OSS 文件存储接口改造

OSS 提供两种访问方式：SDK 和 API 接口。目前已支持 SDK 的开发语言有：Java、Python、Android、iOS、PHP、.NET、NodeJS、C SDK 开发包。如使用其他语言，需要用户自己封装 API 接口。

对于原来采用文件存储服务器方式存储文件、图片信息的程序，需要将原访问接口改造为 OSS 官方提供的标准 SDK 或 API 接口。

对于原来采用关系数据库方式存储文件、图片信息的程序，需要修改关系数据库表结构和程序的数据读写代码。改造关系数据库表结构时，需要将原 BLOB 类型修改为 varchar 类型，存储文件对象所在 OSS 的地址。

SDK 文档资料请参考如下资源：https://help.aliyun.com/document_detail/52834.html?spm=5176.doc31883.6.655.MBW02。

API 文档资料请参考如下资源：https://help.aliyun.com/document_detail/31947.html?spm=5176.doc52834.6.836.8SADUD。

3.4.4 大数据实时计算应用改造

很多情况下，传统行业一个业务系统对应后台一个 Oracle 数据库，Oracle 数据库基本支持全部业务，既包含在线事务型的业务，也包含实时分析类业务，同时还有离线处理类

型的业务。当实时分析、离线处理的业务和在线事务型业务叠加的时候，往往会导致 Oracle 数据库系统不堪重负，使在线业务系统性能出现严重问题，甚至无法使用。

对于导致性能压力过大的实时分析处理类的功能，建议从原有在线业务系统中剥离开。AnalyticDB 作为传统关系型数据库的补充，将部分分析处理类的业务迁移至 AnalyticDB 上执行。对于离线处理类的业务改造将在下一节阐述。

目前适合使用 AnalyticDB 的业务需符合如下特征：

- ❑ 不要求强一致性，可接受最终一致性的业务场景。
- ❑ 对数据更新频率要求不高，可以接受分钟级及以上数据写入延时。
- ❑ 需要计算的数据量单表至少在百万量级，最高可到千亿量级。
- ❑ 对计算速度相对要求苛刻，可满足响应时间在 1s 至 10s 间业务场景的要求。
- ❑ 支持 SQL 语句访问，暂不对外支持复杂的计算，如 MPI、MapReduce、BSP 等计算模型接口。
- ❑ 单个数据表访问，不需要进行多表之间的关联查询，主要是单个大表的上卷、下钻、过滤等多维分析场景。
- ❑ 数据分析类业务，以一个大的事实表为中心，多个较小的维度表和事实表进行关联查询，然后进行上卷、下钻、过滤等多维分析场景。

1. AnalyticDB 库表设计

1) **表类型**：AnalyticDB 库表包括维度表和事实表两类。其中，维度表一般是指存储配置和元信息的表，数据量较小。事实表一般是指系统中数据量为千万级以上的表。

2) **表组**：在 AnalyticDB 中，每一个表都必须属于一个表组。设计时需要充分考虑业务使用情况，将需要关联查询的表划分到同一个表组，同一个表组内，一级分区数量必须一致。

3) **表分区**：AnalyticDB 分区支持 HASH 分区和 LIST 分区，并支持两级分区。事实表必须要指定分区和表组，维度表不需要指定分区但要指定表组。选择一级分区字段时，要尽可能保证数据分布均匀，单个分区的数据量在 1GB 以内或 1000w 级以下。Hash 分区是根据导入数据时已有的一列的内容进行散列后进行分区的，因此目前多张表进行 Join 时，Join 列必须包含分区列，并且表的 Hash 分区数必须一致。另外，采用 Hash 分区的数据表，在数据装载时会全量覆盖历史数据的。List 分区则是根据进行导入操作时指定的分区列值来进行分区的。

事实表的分区设计有两种类型：①只采用一级分区，采用 Hash 分区方式；②采用二级分区，一级分区为 Hash 分区，二级分区采用 List 分区。第一种方式是 AnalyticDB 最常用的分区设计方式，适用于大部分情况。第二种方式中，一级分区使用 Hash 分区，二级分区采用 List 分区。二级 List 分区可支持数据增量导入，如每天新增一个二级 List 分区存储业务新增数据。

4) 数据类型：AnalyticDB 数据类型是 MySQL 数据类型的子集。如表 3-1 所示。

表 3-1 AnalyticDB 与 MySQL 数据类型对照表

AnlyticDB 数据类型	MySQL 数据类型	差异
boolean	bool	一致
tinyint	tinyint	一致
smallint	smallint	一致
int	int	一致
bigint	bigint	一致
float	float[(m	分析型数据库不支持自定义 m 和 d，而 MySQL 可以
double	double[(m	分析型数据库不支持自定义 m 和 d，而 MySQL 可以
varchar	varchar	一致
date	date	一致
timestamp	timestamp	分析型数据库只支持到精确到毫秒，而 MySQL 是可以确定定经度的
multivalue	-	分析型数据库特有的，MySQL 无此类型

注：分析型数据库所有的数据类型都不支持 unsigned，下表不包含该差异

2. AnalyticDB 数据同步和集成

阿里云提供了多种数据同步和集成工具，如 CDP、DTS，可支持从 RDS、Oracle、OSS、文本文件等数据源实时或定期同步到 AnalyticDB 数据库。DTS 数据传输产品可支持从 RDS（MySQL）实时同步数据到 AnalyticDB 数据库，数据延时在分钟级别。CDP 云上数据同步管道可支持关系数据库到阿里云数据存储产品、阿里云数据存储产品之间的定期批量数据同步。同时 AnalyticDB 提供 Web 化的管理控制台，可将 MaxCompute 数据批量导入到 AnalyticDB，以及提供 SQL 接口支持应用程序向 AnalyticDB 写入数据。

3. 使用 AnalyticDB 的注意事项

❑ 表限制

AnalyticDB 数据以二维表方式存储，一般可分为维度表和事实表。

维度表主要用于存储数据量较小，且修改不频繁的业务数据，如 meta 类数据。一般建议维度表数据行数小于 1000 万且占用存储空间小于 1GB。AnalyticDB 维度表不支持分区。维度表和事实表关联查询时，关联列需要在数据导入前添加哈希索引，否则无法进行表关联。

事实表一般建议存储数据量较大，且更新频繁的业务数据，如消费记录数据。事实表一般比较大、数据行在千万级以上。事实表可支持分区，分区列可支持日期或数值类型。使用事实表时需要注意，创建事实表必须指定所属数据库和表组。

事实表的限制如下：

- 一张事实表至少有一级 Hash 分区并且分区数不能小于 8 个。
- 一个事实表组最多可以创建 256 个事实表。
- 一个事实表最多不能超过 1024 个列。

❑ 数据导入限制

AnalyticDB 提供多种数据批量导入的功能, 执行数据导入操作前需要进行相应的授权操作。例如, 数据来源在 MaxCompute 上, 则需要在 MaxCompute 上对特定账号授予源表的 describe 和 select 权限。

AnalyticDB 目前仅允许操作者导入自身为 Project Owner 的 MaxCompute Project 中的数据。如果不满足这些条件, 发起导入命令时会报 can not load table meta 错误。

❑ 使用限制

为了更高效地进行表关联, AnalyticDB 中两个事实表进行关联查询必须满足以下充要条件:

- 两张表在一个表组。
- 两张表的 Join Key 是 Hash 分区列。
- 两张表的 Hash 分区数必须一致, 否则 Join 结果不准确。
- 两张表的 Join Key 至少有一列建立了 HashMap 索引, 推荐建立在数据量较小的一侧。
- 维度表参与关联查询, 只需要符合上述第 4 点即可。

上述内容为 AnalyticDB 对多表关联查询的限制。另外, AnalyticDB 还会对通过 SELECT 语句进行查询的结果返回行数设置限制, 默认只返回 10000 行。

3.4.5 大数据离线分析应用改造

对于用户大数据离线分析类应用, 建议对其进行改造, 以已匹配阿里云 MaxCompute 大数据计算服务。

大数据计算服务是一种快速、完全托管的海量数据仓库解决方案。MaxCompute 向用户提供了完善的数据导入方案以及多种经典的分布式计算模型, 能够更快速地解决用户海量数据计算问题。MaxCompute 主要服务于批量结构化数据的存储和计算, 可以提供海量数据仓库的解决方案以及针对大数据的分析建模服务。

1. 使用场景

MaxCompute 主要用于离散数据存储和批量计算场景, 不适用于实时存储和分析场景。一般运行在 MaxCompute 之上的业务系统对数据时效性要求低, 通常是 T+1 的系统, 同时 MaxCompute 可支持 TB/PB 级数据的离线分析和处理。使用 MaxCompute 的一般做法是, 数据从业务系统抽取到 MaxCompute 后, 经过 MaxCompute 离线分析处理后, 将分析处理后的结果回写到业务系统, 业务系统直接查询和访问 MaxCompute 分析处理

后的结果。

目前有大量用户的数据仓库类应用都是基于 Oracle 关系数据库建设的，当用户数据量达到 TB 级特别是 10TB 以上后，大规模离线数据计算场景 Oracle 数据库计算性能已无法满足用户的业务需求。下面将以基于 Oracle 数据库构建的大数据分析应用为例，介绍如何从 Oracle 迁移到阿里云 MaxCompute 服务。

2. MaxCompute 结构与改造

(1) MaxCompute 表设计

MaxCompute 的存储结构与关系型数据库有本质差异，需要根据 Oracle 原始表结构，对其在 MaxCompute 中的存储方式与结构进行设计。由于其不是一一对应的关系，故需要对每个数据表进行逐一的核对后，方可最终确定 MaxCompute 的数据结构。

数据类型转换规则如表 3-2 所示。

表 3-2 Oracle 到 MaxCompute 数据类型转换表

Oracle 数据类型	MaxCompute 数据类型	备注
BLOB	STRING	将 BLOB 转换为 OSS 地址
INTEGER	BIGINT	
NVARCHAR2	STRING	
NUMBER	BIGINT	19 位以下整型数据
NUMBER	DECIMAL	19 位以上
CHAR	STRING	
DATE	DATETIME	
VARCHAR2	STRING	
CLOB	STRING	将 CLOB 转换为 OSS 地址
TIMESTAMP	DATETIME	
FLOAT	BIGINT	
NCHAR	STRING	

(2) MaxCompute 表属性设计

MaxCompute 表属性设计主要包含表命名规范、表注释、表数据生命周期、是否开启极限存储等方案。

在表命名规则方面，建议和业务系统数据库名、表名保持一致，如采用业务系统库名_业务系统表名方式。表注释主要用于描述表的功能和作用，注释内容要求不超过 1024 字节。

MaxCompute 提供数据生命周期管理功能，方便用户释放存储空间，简化回收数据空间的流程。用户可在创建表时通过 lifecycle 属性指定数据生命周期，生命周期时间必须设置为正整数，单位是天。数据表是否开启极限存储功能主要适用于按天分区的全量快照表，且不同分区的数据重复度高的场景。

(3) MaxCompute 分区设计

MaxCompute 主要用于离线存储、批量计算和分析，数据源于在线业务系统数据库。一般采用批量同步技术，按指定周期将在线业务数据库数据同步到 MaxCompute 进行存储和分析。为方便存储和分析，会使用 MaxCompute 分区表存储数据，表分区规则建议按时间划分分区，表分区键及定义为：dt string。MaxCompute 分区分为静态分区和动态分区，可以做多个分区层级。一个表允许的分区个数支持按照具体的 project 配置，默认值为 60 000 个。

静态分区在插入数据前已通过添加分区方式建立。动态分区在使用 insert overwrite 插入到一张分区表时，可以在分区中指定一个分区列名，但不给出值。相应的，在 select 子句中的对应列来提供分区的值。动态分区的限制如下：

- ❑ 目前，在使用动态分区功能的 SQL 中，在分布式环境下，单个进程最多只能输出 512 个动态分区，否则引发运行时异常。
- ❑ 在现阶段，任意动态分区 SQL 不可以生成超过 2000 个动态分区，否则会引发运行时异常。
- ❑ 动态生成的分区值不能为 NULL，否则会引发异常。
- ❑ 如果目标表有多级分区，在运行 insert 语句时允许指定部分分区为静态，但是静态分区必须是高级分区。

注意，在表中出现的分区层次不能超过 6 级。

3. SQL 改造

MaxCompute 只能以表的形式存储数据，并对外提供了 SQL 查询功能。用户可以将 MaxCompute 作为传统的数据库软件操作，但其却能处理 TB、PB 级别的海量数据。MaxCompute SQL 与 Hive SQL 高度兼容，但 MaxCompute 不支持事务、索引及 Update/Delete 等操作，同时 MaxCompute 的 SQL 语法与 Oracle、MySQL 有一定差别，用户无法将其他数据库中的 SQL 语句无缝迁移到 MaxCompute 上来。因此，需要结合 MaxCompute 语法进行 SQL 语句改写，MaxCompute SQL 语法请见官方文档：

https://help.aliyun.com/document_detail/27862.html?spm=5176.doc27863.6.603.o1grir

https://help.aliyun.com/document_detail/27863.html?spm=5176.doc51823.6.604.SJatSw

https://help.aliyun.com/document_detail/48950.html?spm=5176.doc27860.6.605.fQYbxR

4. 数据同步和更新

使用 MaxCompute 服务后，需要配置同步任务从在线业务系统定时或实时同步更新 MaxCompute 数据。可以使用阿里云 BASE 平台配置数据同步任务完成 MaxCompute 数据同步，再通过 MaxCompute SQL 实现 MaxCompute 历史数据的更新操作。

5. MaxCompute 的注意事项

使用 MaxCompute 时，有一些注意事项，如表 3-3 所示。

表 3-3 使用 MaxCompute 的注意事项

select 语句	• 当使用 select 语句屏显时，目前最多只能显示 1000 行结果。当 select 作为子句时，无此限制，select 子句会将全部结果返回给上层查询 • IN、NOT IN 内子查询返回行数限制为 1000 • MaxCompute SQL 的 where 子句不支持 between 条件查询 • order by 必须与 limit 共同使用 • 在使用 order by 排序时，NULL 会被认为比任何值都小，这个行为与 MySQL 一致，但是与 Oracle 不一致 • sort by：局部排序，语句前必须加 distribute by • order by 不和 distribute by/sort by 共用，同时 group by 也不和 distribute by/sort by 共用，必须使用 select 的输出列别名 • order by/sort by/distribute by 的 key 必须是 select 语句的输出列，即列的别名
子查询	子查询必须要有别名
union all	• MaxCompute SQL 不支持两个顶级的查询结果合并，要改写为一个子查询的形式 • MaxCompute 最多允许 128 路 union all，超过此限制报语法错误
join	• 支持多路连接，但不支持笛卡儿积，即无 on 条件的链接 • 连接条件，只允许 and 连接的等值条件，并且最多支持 16 路 join 操作
mapjoin hint	• left outer join 的左表必须是大表；right outer join 的右表必须是大表；inner join 左表或右表均可以作为大表 • full outer join 不能使用 mapjoin • mapjoin 支持小表为子查询 • 使用 mapjoin 时需要引用小表或是子查询时，需要引用别名 • 在 mapjoin 中，可以使用不等值连接或者使用 or 连接多个条件 • 目前 MaxCompute 在 mapjoin 中最多支持指定 6 张小表，否则报语法错误 • 如果使用 mapjoin，则所有小表占用的内存总和不得超过 512MB
Lateral View	Lateral view 通常和 split、explode 等 UDTF 一起封装使用，它能够将一行数据拆成多行数据，在此基础上可以对拆分后的数据进行聚合