



SaaS模式与技术架构

C 目录

CONTENTS

1

SaaS简介

2

SaaS与传统软件模式

3

SaaS、SOA、云计算

4

SaaS应用架构

5

SaaS面临的挑战

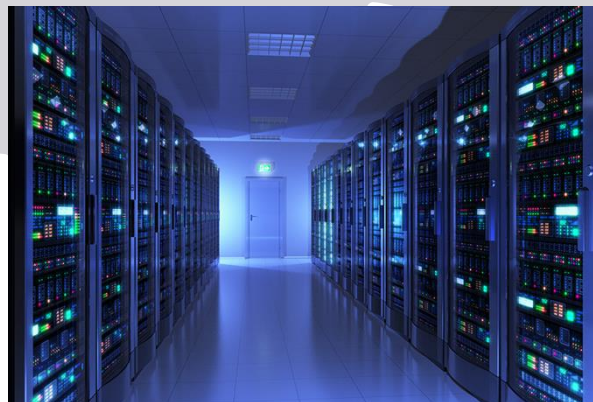
SaaS简介



软件是一种产品

软盘
光盘

...



软件是托管的应用

服务器



软件即服务

平台

SaaS简介

起源于1999年，客户关系管理（CRM）软件服务商Salesforce最早实现并运用于CRM行业。

SaaS（软件即服务）指的是软件的应用交付模式。在这种模式下，软件供应商开发了基于浏览器的软件应用和托管，并让客户运行这些应用程序。（或独立运营或给予第三方），在这种模式下，客户不必支付软件的所有费，只付使用费。他们通过可访问的API使用这种服务，并且通过Web Service或者REST来写入。SaaS已经成为行业中推崇的术语，代替了先前的ASP和On-Demand。Web应用程序不等于SaaS，SaaS不一定用到云计算。SaaS是一种以Internet交付的，依使用付费，使用者不需维护软硬件的商业模式。



SaaS简介



从技术方面来看

企业无需再配备IT方面的专业技术人员，同时又能得到最新的技术应用，满足企业对信息管理的需求

SaaS简介



从投资方面来看

企业只以相对低廉的“月费”方式投资，不用一次性投资到位，不占用过多的营运资金，从而缓解企业资金不足的压力；不用考虑成本折旧问题，并能及时获得最新硬件平台及最佳解决方案。

SaaS简介



从维护和管理方面来看

由于企业采取租用的方式来进行物流业务管理，不需要专门的维护和管理人员，也不需要为维护和管理人员支付额外费用。很大程度上缓解企业在人力、财力上的压力，使其能够集中资金对核心业务进行有效的运营。

SaaS、SOA、云计算

SOA

SOA是一种面向服务的架构，一种软件架构设计的模型和方法论。他可以看做模块化组件，每个模块都可以实现独立功能，而不同模块之间的结合则可以提供不同的服务，模块之间的接口遵循统一的标准，可以实现低成本的重构和重组。

SaaS

SaaS是一种基于互联网面向服务的新型软件交付模式。

云计算

云计算（Cloud Computing），是一种基于互联网的计算方式，通过这种方式共享的软件资源和信息，可以按需提供给计算机和其他设备。

层服务：IaaS、PaaS、SaaS。

SaaS侧重于运营和交付，SOA侧重于平台架构。SaaS是云计算的最终价值交付运营模式，SOA是实现云计算开发架构的基础理念之一，两者都包含了云计算的某些重要特征，但是他们都不能称之为云计算的全部。

SaaS、SOA、云计算



SaaS与传统软件模式

对比项	SaaS	传统软件
部署模式	采用托管模式，部署在统一服务器上并可方便支持移动通讯办公、异地灾备等	采用系统集成方式部署在服务器上，只对本单位使用者负责本单位
维护管理	用户不需要专业的IT维护人员	必须有专业的IT技术人员做维护工作
数据处理能力	满足日益增长的计算能力需求	软件需要定时、及时升级或重新开发换代
数据管控	由运营商提供数据的维护和集中管理，用户数据有被攻击、篡改、丢失的风险，必须采用相应的技术手段认证接触数据者的身份并可追溯	本单位人员管理，人员风险相对小
购买方式	租用方式，可免费试用	一次性购买并全费支付，在以后的维护中还能追加成本
系统风险	可灵活定制、快速开发上线，周期短，项目本身风险低	系统实施速度慢、周期长

SAAS架构



火龙果·整理
uml.org.cn

Dharmesh Shah撰写了有关“软件作为服务”即SaaS的架构成熟度模型的文章。之前Gianpaolo Carraro曾提出可伸缩性、多租户 (multi-tenancy) 与通过配置进行定制 (customization through configuration) 是SaaS架构应满足的要求, Dharmesh从这一点出发, 提出了关于SaaS架构成熟度模型的5个级别——从“混乱”到“乌托邦”, 并针对每个级别背后的经济考虑提出了自己的想法。

第0级 (混乱) : 每次新增一个客户, 都会新增软件的一个实例。

第1级 (受控的混乱) : 所有客户都运行在软件的同一个版本上, 而且任何的定制化都通过修改配置来实现。

第2级 (多租户[multi-tenant]、高层建筑[Highrise]) : 所有的客户都已经可以在软件的同一个版本上运行了, 而且他们都在同一个“实例”上运行。

第3级 (多租户, 扩建[Build-Out]) : 此时你已经拥有了多租户、单一版本的软件模型。不过你还是可以通过硬件扩展 (scale-out) 的方式来进行扩充。

第4级 (乌托邦) : 如同第3级, 除非你可以找出有效的方式, 以在不同的“实例”上运行不同版本的软件。

SAAS架构

成熟度分级

级别	可配置性	高性能	可收缩
Level1 定制开发	N	N	N
Level2 可配置	Y	N	N
Level3 高性能的多租户架构	Y	Y	N
Level4 可伸缩性的多租户架构	Y	Y	Y

SAAS架构

多租户模式下的数据隔离

模式	隔离级别	共享级别	安全级别	成本
独立数据库	高	底	高	高
共享数据库、 隔离表架构	中	中	中	中
共享数据库、 共享表、隔 离数据架构	底	高	低	底

SAAS架构

数据库层性能优化



Index

The diagram features a large orange circle with the word 'Index' inside. The background is a light gray world map.

建立适合的索引



Table

The diagram features a large dark blue circle with the word 'Table' inside. The background is a light gray world map.

消除大数据表连接
冗余字段



SQL

The diagram features a large orange circle with the word 'SQL' inside. The background is a light gray world map.

避免复杂SQL

一般情况下、应用服务器层的水平扩展更容易实现，而数据库层扩展则比较困难

SAAS架构

应用层性能优化



cache

使用适当的Cache是提升性能的重要措施

需要Cache的数据应该是读多写少
Cache是易失的，应增加一层逻辑。访问不到时访问数据库，并更新Cache。
MemCache是由Danga Interactive开发的高性能分布式的内存对象缓存系统。

SAAS架构

应用层性能优化

异步

表现层异步，一般用Ajax实现

.net 异步调用分为两大类“等待”和“回调”

第一种方法：BeginInvoke EndInvoke方法，属于“等待”类。

第二种方法：WaitOne。同样属于“等待”类。

第三种方法：轮询。也是属于“等待”类。

第四种方法：回调。当然属于“回调”类。

SAAS架构

数据配置方案

定制字段

根据客户的需求在数据表上增加相应的定制字段来保存扩展数据。对于SAAS应用来说，定制的字段的数量多如牛毛，显示不是解决SAAS应用下数据可配置的理想方案。

预分字段

将所有扩展数据转换成字符串进行存储，对于租户用各字段保存的真实数据类型，由租户配置并作为配置元数据进行管理。系统可以根据元数据配置信息转换成真实的数据类型。

SAAS架构

数据配置方案

名值对称

可以将扩展数据的保存与原数据表分离，用一张统一的扩展数据表保存

租户

id	TenantID	Name
112	40	Joy

扩展字段

ConfigID	TenantID	Table	Content	DataType
20	40	custom	Salary	int

数值

ExtID	Table	DataID	ConfigID	ExtVlue
1002	custom	112	20	3000

SAAS架构

功能配置方案

功能分解

每个功能都是有价值的，每个功能都是不可再分的，功能间不相互重叠，功能间不循环依赖，整个系统是完整的。

功能定义及依赖

所谓功能依赖是指一个功能在没有另外功能情况下不能使用。

功能包设计

根据用户的类型和系统的业务逻辑，综合考虑用户的使用场景，将原子功能进行组合成功能包。

销售包设计

功能包不能完全独立使用，还需要按不同的商业意图构造事宜于用户使用的销售包。

功能使用校验

在原子功能使用前，对当前用户是否可以使用该原子功能进行校验。

SAAS架构

界面配置方案

系统菜单可配置

一个租户一套菜单，
一个菜单关联一个原子功能、组织成树状结构、同级菜单之间存在顺序问题。

页面元素可配置

按钮、文本、提示、
校验等等

SAAS架构

可伸缩的SAAS应用架

伸缩性 (Scalable) 的概念

Scalable有时被翻译成“可扩展”，容易被误解为功能扩展，实际Scalable更多强调性能、容量方面可扩展。

Scalable伸缩要求：系统规模/容量小时。系统可说下，大时可以增加，而不需要更改系统整体架构，最理想涨停是随着用户数的增加，系统架构不用做调整，而仅需要增加/增强硬件设备。

可伸缩的最简单方式就是Scale up，译为垂直扩展，也就是增强硬件设备，例如一个普通Server不行，换成小型机。此种扩展面临高成本问题。

通常强调的应用架构可伸缩，一般是指Scale out，即水平扩展，例如Google搜索引擎是由数千万台普通PC Server构成的。

SAAS架构

可伸缩的SAAS应用架

应用服务器层的水平扩展

应用服务器层的负载均衡，是实现应用服务器层水平扩展的最主要手段

基于硬件负载均衡设备实现负载均衡，如F5设备。

基于软件的方式实现负载均衡，例如我们通过WebLogic Server集群（Cluster）可以实现服务网络的负载均衡、失败转移。

SAAS架构

可伸缩的SAAS应用架

数据库层的水平扩展

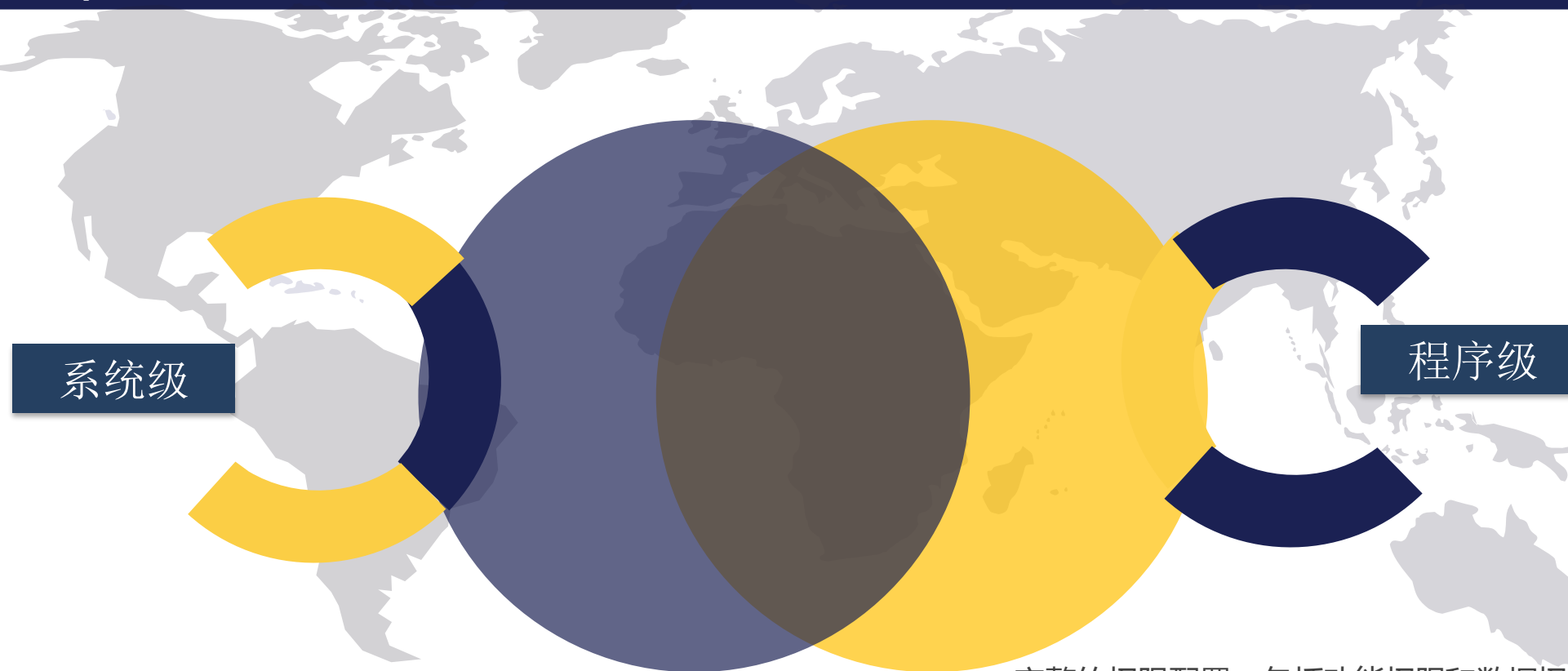
相对于应用服务器层的水平扩展，数据库层的水平扩展更难实现。

实现数据库层的水平扩展有多种方式。

- 1、数据库的垂直切分：将不同的功能模块所涉及到的表划分到不同的物理数据库中，从而将对这些表的访问压力分担到不同的物理数据库；
- 2、数据库的读/写分离：同一数据库在多个物理服务器上具有多份Copy，彼此同步，写操作都统一到一个主服务器上，读操作则分担到多台从服务器上；

SAAS架构

SAAS安全性



系统级

程序级

使用HTTPS协议以SSL (Security Socket Layer) 交换数据, 增强通信安全;
通过数字签名防止传输过程篡改;
对用户身份识别的UserToken使用DES算法数据加密;
业务数据定时自动备份。

完整的权限配置, 包括功能权限和数据权限;
客户端输入校验, 防止JS攻击、XSS攻击、SQL注入等;
辅助安全设计, 比如密码控件、图片验证码、手机确认码等。

SAAS架构

SAAS安全性

安全压倒一切。大多数用户只是问问SaaS厂商是不是采用了安全套接层(SSL)技术，而安全性涵盖的不仅仅只有这个方面。要向潜在的SaaS厂商询问下列问题：

- 放置服务器的数据中心有没有24×7全天候的物理安全措施？
- 数据中心有没有得到保护(保安是不是24小时在周围至少巡视一次)？
- 谁有权访问这些服务器(只有内部员工可以访问，还是承包商也可以访问？)
- 有没有日志记录谁何时进入、何时离开？如果有日志，那么隔多长时间审计这些日志？
- 应用程序有没有使用基于行业标准的128位加密技术？
- 如果多个客户使用的应用程序放在同一台服务器上，那么它们有没有采用逻辑或物理分隔从而确保你的数据不被未经授权的人所看到？
- SaaS厂商中可以访问你企业数据的工作人员有没有经过犯罪背景调查？知道被定罪的重罪犯是不能访问你企业那些敏感的个人数据，这很重要。
- 厂商有没有正规的业务连续性方案(BCP)？对方愿不愿意与你共享该方案、它能消除你的担忧吗？

SAAS面临的挑战



依赖互联网

政府和商业机构重视互联网集成设施的建设，互联网的接入方式越来越多，带快越来越大，数控越来越快，可靠性、稳定性、抗灾性越来越强。



数据安全性

T高可靠性存储环境，制定数据备份、容灾策略，规范和提高数据存储服务的质量和和安全指标。



数据保密

T软硬件保密措施，加强SaaS软件上的信用建设



THANK YOU FOR WATCHING