

白皮书

服务虚拟化解决方案的主要功能 | 2013 年 1 月

借助服务虚拟化 消除软件开发和 测试的限制

John Michelsen
CTO/CA Technologies

agility
made possible™



目录

执行摘要	3
-------------	----------

第 1 部分： 创建“真实”的环境	4
------------------------------	----------

第 2 部分： 实现并行开发和测试	5
------------------------------	----------

第 3 部分： 处理范围外依存关系的测试数据	6
-----------------------------------	----------

第 4 部分： 支持异构技术和平台	8
------------------------------	----------

第 5 部分： 结束语	10
------------------------	-----------

第 6 部分： 作者简介	10
-------------------------	-----------

执行摘要

挑战

所有依赖于 IT 的大公司都必须面临系统的约束，例如大型机、开发中的组件以及敏感数据源导致项目延迟。团队为了进行测试，不可避免地会尝试在测试实验室中复制完整的环境，或者通过对他们自己的版本进行编码来“模拟”响应系统。这已成为一项昂贵又耗时的的工作。

机遇

实际上，服务虚拟化 (SV) 是模拟开发和测试环境并将其用作存根这一做法的产品化，具有加快开发速度所需的足够的可行性和上下文，同时**在生命周期中将测试左移**，因此集成和发布流程得以更快实现，并且质量更高、风险更小。

优势

接下来将介绍您的组织可从服务虚拟化解决方案中获得的一系列功能，这些功能使您的扩展团队能够以更低的成本和风险将应用快速投入市场。这些功能包括：

为开发提供一个更加真实的环境	更高质量的开发和更为有效的回归/系统测试
实现并行开发和测试	缩短周期时间、更早发现缺陷、高效利用资源
将范围外系统的测试数据虚拟化	设置/拆卸更快速、测试自动化更稳定
实现高性能环境	提高了性能测试的可行性和数量，并显著了降低成本

由于没有哪个应用开发和测试环境是一座孤岛，因此服务虚拟化解决方案需要为团队可能拥有的特别工具提供一个“不依赖于供应商”的基础。服务虚拟化会提供与现有的应用周期解决方案（例如测试管理 (TM)、缺陷管理/问题跟踪以及环境中现有的领先硬件和测试实验室虚拟化产品）共同起作用的目标环境。

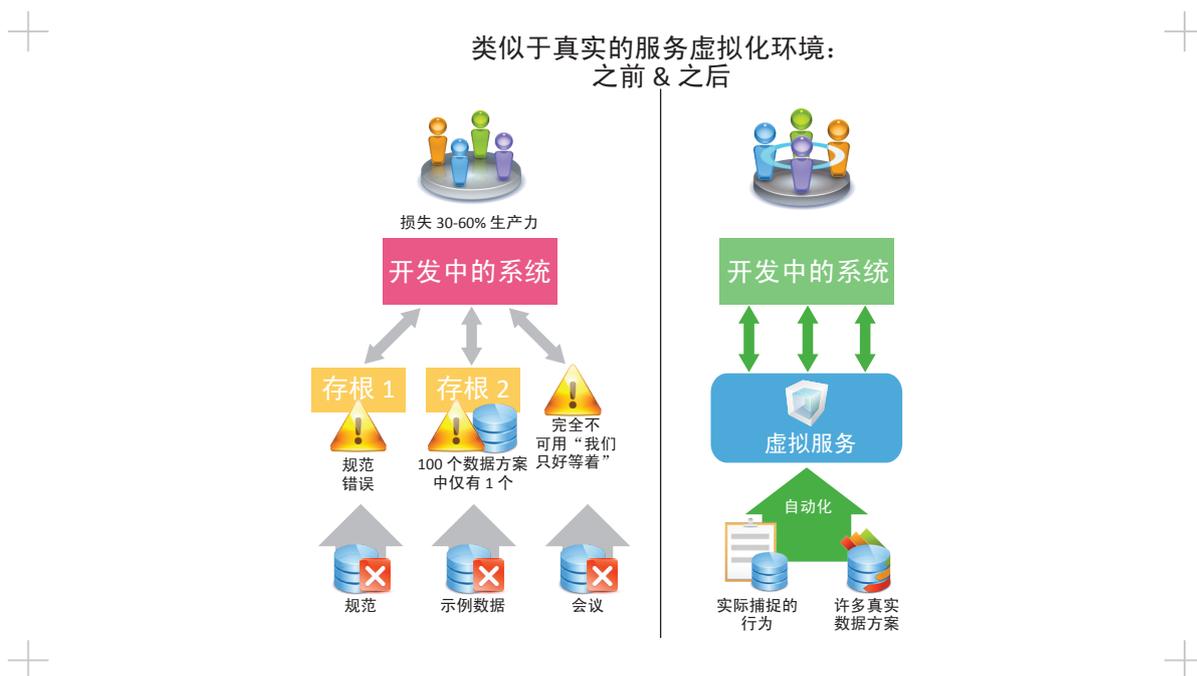
第 1 部分：创建“真实”的环境 在开发和测试环境中模拟从属系统

由于应用开发趋向于采用更为复合、面向服务的体系结构方法，因此团队必须在他们的开发和测试环境中模拟越来越广泛的上游和下游系统。应该在任何存在依赖关系的层上或层之间应用服务虚拟化，以提供最为真实或尽可能“真实”的环境。

在传统的方式中，团队试图仅通过将下一个下游系统用作“存根”来推动自己的组件开发。

例如，假设我正在开发一款 Web UI，并为来自下一层的几个预期响应建立了存根（如 Web 服务）。然后，Web 服务开发人员可能会剔除他们的基础 ESB 层，或者尝试模拟一些来自 Web UI 的用户请求。遗憾的是，要封装企业软件体系结构中的多种类型的连接和数据，靠这个手工流程是远远不够的，而且如果还没有已编码的 UI（如下所示），这个手工流程可能完全不可用。

图 A.



或者，当团队处理作为虚拟服务进行捕捉的真实数据方案和真实行为时，生产力水平会提高，因为相较于必须进行手工编码和维持的存根组，通过这种方式创建的环境更为真实，也更接近当前。

所以，实现“真实”环境的关键方法是虚拟服务创建和数据维护的自动化。这使得开发团队由于使用真实的虚拟实验室环境（即用户界面不完整）而提高了生产力，与此同时，还缩短了创建或修改过时存根的时间。

预期优势:

- 即使界面系统不可用也能够启动开发
- 缩短执行测试的周期时间
- 由于减少了对其他应用数据的依赖并缩短了可用的测试时间，因此可以扩大测试覆盖范围
- 以更少的工作改善单元测试
- 测试覆盖范围和回归测试的增加改善了代码质量
- 能够以较少的维护工作更快地构建模拟器

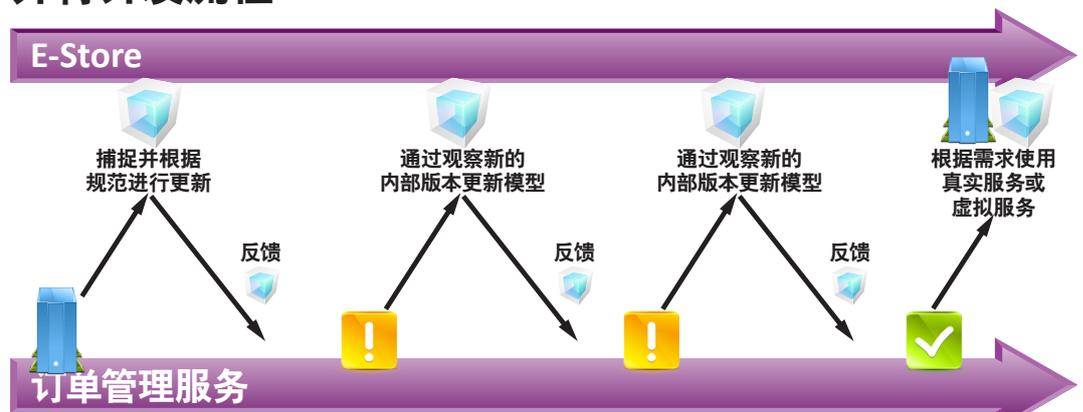
第 2 部分：实现并行开发和测试 开发和测试团队同时工作

服务虚拟化解决方案的第二大主要功能是实现并行开发和测试。当开发和测试团队同时工作时，整个软件周期的功效和效率将达到一个全新的水平。向组织交付的新解决方案可以创造更大的价值。

在并行开发和测试中，虚拟服务以共生方式充当开发中的系统与测试中的系统之间的“中间”资产。在下面的示例中，一个团队正在开发一个订单管理服务 (OMS)，与此同时，顶部的团队正在开发和测试一个 e-store 应用。

图 B.

并行开发流程



将从现有的 OMS 系统捕捉一个虚拟服务，作为 e-store 测试活动的初始后端。然后，随着测试的继续，e-store 团队可以传回任何意外的或新的响应要求，作为“反馈”虚拟服务请求，这实质上将成为下一组开发要求。随着每个新的内部版本的产生，会发生新一轮虚拟服务模型更新，而且反馈也越来越快，并行开发和测试周期将不断加速。

理想的并行开发解决方案允许团队针对真实服务执行，这些真实服务是可用的、功能强劲并已进行数据同步。如果团队没有可正确支持组件的服务，则他们可以立即转回到虚拟服务。这一通过虚拟服务在纯虚拟下游系统或真实系统之间轻松切换的能力，对于实现强劲的并行开发功能来说是十分有用的资产，因为您知道，如果新的内部版本出现中断或者需要新的数据方案，您总是可以返回。

您生成和更新软件实验室环境的做法是否适合现实情况？实际上，服务虚拟化的并行性实现了我们所期待的复杂软件环境的敏捷性 — 在紧密的循环发布中，使开发和测试周期更加符合业务发布目标。

预期优势：

- 测试和开发周期的速度不断增长
- 通过根据测试结果和业务要求不断进行集成和构建，实现敏捷循环的真正响应能力
- 减少版本控制的负担，与现有的 ALM 和开发管理工具一起使用以使其更为有效
- 在进入生产环境之前提高问题受理和解决率
- 将交付功能点的速度提高达 60%，并且规格的质量和精确性也更高

第 3 部分：处理范围外依存关系的测试数据 下游数据缺失将不再是一个问题

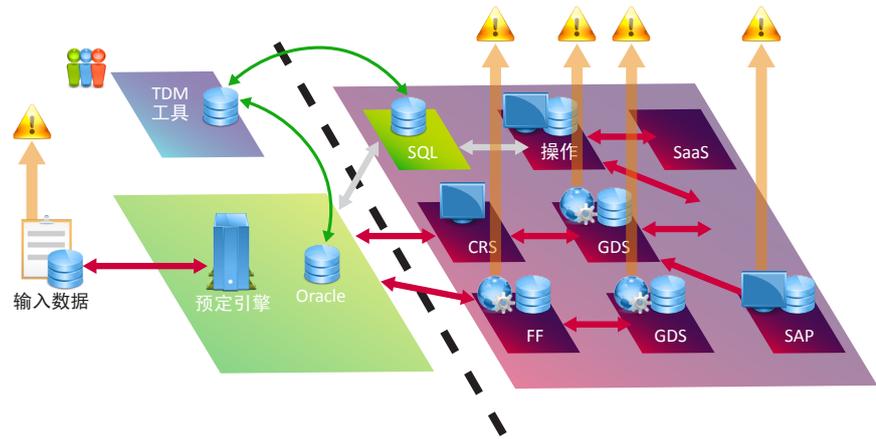
每个需要为他们的活动开通实验室的团队都拥有一些认为是范围内的系统和关联数据，以及一些认为是范围外的系统和关联数据。范围内的系统即他们直接在其中执行开发变更或测试的系统。范围外的系统即支持范围内系统所需的系统，但事实上并非开发或测试活动的主体。这视为一种依赖关系。这些系统是必要的，但并非开发或测试活动的主体。

在此方案中，团队正在开发和测试预定引擎，他们尝试汇集所需的下游和用户输入数据方案，但是他们没有使用整个领域的权限。

最常见的解决方案是习惯做法 TDM，即直接从范围内系统导入数据，然后通过尝试写入代码并导入几行数据来代表范围外依赖关系的预期响应，从而模拟范围外系统或将其用作“存根”。这些存根本质上很脆弱，开发人员在模拟他们预期的基本功能或响应方案时均采取最便捷的方法。但是，随着当今分布式软件的复杂性不断增加，对有用的存根进行手工编码和维护已经变得过于昂贵。

图 C.

TDM 未覆盖超出范围的数据



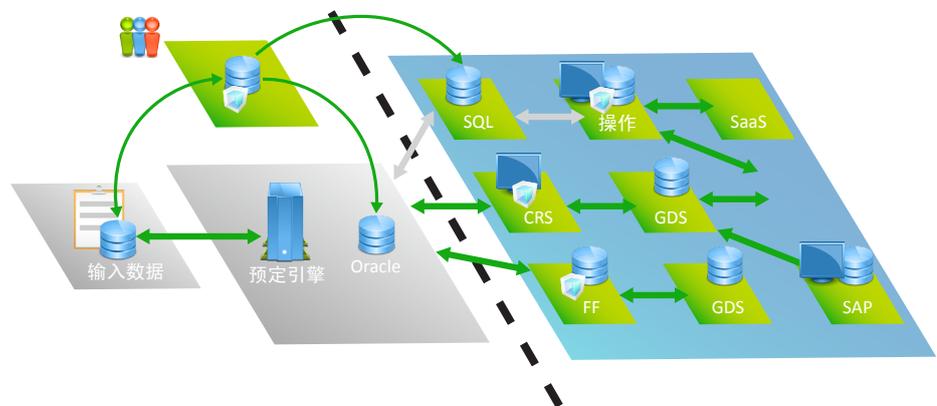
是真实的数据还是虚拟的数据？

我们需要一种以足够的智能模拟范围外系统的行为的方式，以便范围内系统认为它是在与真实的系统对话，但事实上并非如此。

服务虚拟化通过自动执行捕捉范围外相关下游方案的行为，使得在范围内系统后面的所有缺失数据不再成为问题。

图 D.

将范围外的测试数据虚拟化



这里显示的 vTDM 的最终状态示例看起来有些过于简单，但是这正是您的测试流程与外部或范围外依赖关系之间的交互过程。借助虚拟模型，您的所有团队可以始终为测试中的系统按需访问关联数据集，这些数据几乎将覆盖无限量的有效数据方案，以支持大容量性能和回归测试需求。

服务虚拟化功能应包括用于将所需的所有系统的提供给开发和测试实验室环境的机制，其中包括开通范围外系统的数据，以及在“有状态”的事务数据随着时间的推移在工作流中经过各个系统时对其进行维护。例如，模拟值的更改过程（例如一个长期事务中的日期或累积量），以便这些类型的复杂工作流可以在必要的详细级别上进行验证。

预期优势：

- 消除由于缺乏访问或当前数据来自范围外系统而导致的延迟
- 全天候为多个测试和开发团队提供有效的测试方案
- 通过在系统中覆盖或更改数据，不再出现测试数据的冲突，也不会再使其他团队的活动无法进行
- 对关键的真实系统的影响极小或没有影响
- 支持“有状态”的事务，以支持跨多个技术层且需要维护日期、客户 ID 等特定值的复杂流程
- 将数据设置和重置所花费的时间缩短达 90%，从而使整个测试周期时间缩短 40-60%

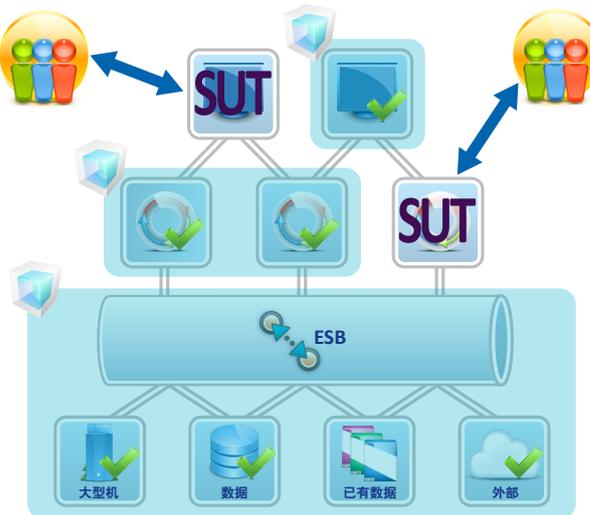
第 4 部分：支持异构技术和平台 创建真正完整的环境

最近我拜访了世界最大银行之一的体系结构团队，我听到了一个惊人的统计数据。这位先生说，他们的硬件资产管理系统声称银行中部署的服务器比银行的员工还要多。每个项目团队都要证明自己的开发、测试、预生产和生产硬件支出费用的正当性。银行中当前正在使用的大多数应用背后至少有四个环境，即使很多情况下要每隔几个月或者几年才会发布一次这些应用的维护或更改。

具有讽刺意味的是，大多数现有预生产环境面临的最严峻挑战居然是从来没有一个完整的系统。一个项目团队可以购买任意数量的服务器，并使用这些服务器来复制团队有权访问的虚拟机上的一些硬件或组件。但是，即使每个团队都分配有他们自己的硬件预算，他们仍然会花费开发周期中的大量时间来等待获得访问权限以及低效地访问共享系统资源。

异构系统在企业 IT 环境中已成常态。所以，应该应用服务虚拟化，以将所有可能影响测试中系统 (SUT) 的依赖关系虚拟化。这包括 Web 通信 (HTTP)、Web 服务 (SOAP/XML)、集成层 & ESB (JMS 等)，以及模拟与基础大型机 (CICS、CORBA 等)、数据库 (JDBC 等) 和第三方服务之间的事务和连接。

图 E.



由于您的软件体系结构中的每个连接点都表示一个潜在的更改点，也代表一个潜在的故障风险，因此，服务虚拟化为团队提供了一个更好的方式“将其他一切内容虚拟化”，从而将他们自己从对异构组件的依赖中解脱出来，这一点非常关键。

使用服务虚拟化，数以百计的预生产实验室可为任何所需环境按需进行基于软件的开通，从而更加简单地管理基础架构。当前未在更改的项目将不再消耗电力、产生热量或占用地面空间。将捕捉和消除依赖关系。团队将按需开通客户信息管理系统的虚拟服务，而无需对大型机分区进行真实系统访问。

预期优势：

- 所有开发和测试团队都能够全天候访问孤立的实验室环境，从而更快地进行交付
- 降低传统预生产基础架构的成本 — 对于大型企业每年可能节省超过 2000 万美元
- 消除访问远程系统的服务成本和费用
- 允许团队在进行传统的用户验收测试活动之前，“将测试左移”并在组件开发和系统集成层进行高质量的构建。将数据设置和重置所花费的时间缩短达 90%，从而使整个测试周期时间缩短 40-60%

另外请注意，异构性适用于团队用于协作的流程层面的工具，而非仅指您所虚拟化的资产。服务虚拟化解决方案为团队可能拥有的特别工具提供了一个“不依赖于供应商”的基础，这一点至关重要。服务虚拟化工具提供与现有的应用周期解决方案（例如测试管理 (TM)、缺陷管理/问题跟踪、测试数据管理 (TDM) 以及环境中现有的领先硬件和测试实验室虚拟化产品）共同起作用的目标环境。

第 5 部分：结束语

迈出正确的第一步

有多少次您带着下面的问题去找开发或测试经理：“您为什么不在您所说的时间交付？”您所听到的只是解释延迟的各种理由：都是因为其他团队，因为他们没有履行承诺和挑战。与我们相关的以及我本人经历的例子有：

- “是的，但是我们本该上个月拿到新订购管理系统的内部版本，但事实上我们上周才拿到。”
- 或者“是的，但是订购管理系统的每个新内部版本都加入了一些我们需要的新内容，但却打断了一些我们在做的其他事情。”
- 或者“是的，但是其他团队给我的订购管理存根中只有一个客户概况，所以我没法构建或测试其他方案，但我要完成这件事却恰恰需要这些。”

最终，只有在其他人完成我们所依赖的“是，但是……”工作之后，我们才能完成我们的工作。与复合应用关联的每个团队都必须自由地按需从基础架构创建自己的实验室。这将不可避免地产生团队内依赖关系。这就是服务虚拟化功能的重要性所在。每个团队都能自由地从下游依赖关系中获取规格并构建下游依赖关系的预期未来状态，甚至不需要看到这些下游依赖关系的第一个内部版本。

服务虚拟化为企业软件开发和测试提供了一个重要平台。在为实现 SV 规划战略时，还要考虑许多其他细节，尤其是组织方面。谁提供解决方案，又是谁使用解决方案？在您的团队和合作伙伴构成的扩展组织中，谁拥有和管理虚拟服务？一个完善的服务虚拟化方法不但要考虑技术细节，还要考虑操作细节，这能最好地保证广泛采用和成功。

第 6 部分

作者简介

John Michelsen, CA Technologies 的 CTO

作为 CA Technologies 的首席技术官，John 负责技术领导和创新，进一步开发公司的技术社区，以及调整其软件战略、体系结构和合作伙伴关系以提供客户价值。John 拥有多项专利，包括以数据库交付的市场领先的发明、分布式计算、虚拟/云管理、多渠道 Web 应用门户和服务虚拟化 (LISA)。

1999 年，John 成立了 ITKO，并从头始构建了 LISA，以优化当今的异构、分布式应用环境。在他的领导下，LISA 用于敏捷开发的平台在广度和深度上都有很大的进步。该公司在 2011 年被 CA Technologies 收购。CA LISA 的产品套件在重塑客户的软件生命周期方面取得了重大成果。如今，它为客户提供 1000% 以上的 ROI，是服务虚拟化市场的领导者。

在 ITKO 之前，John 为 Trilogy 和 Agency.com 上的跨国公司领导 SaaS 和电子商务改革。他还针对美国航空公司、Citibank 和 Xerox 等客户成立了一个专注于分布式、任务关键型应用开发项目的精品定制软件公司。

John 在分别在三一大学和哥伦比亚大学获得了商学学位和计算机科学学位。他撰写过一本有关最佳实践的书：“Service Virtualization: Reality is Overrated”（服务虚拟化：现实被高估）（可在 servicevirtualization.com/book 上找到）。他还在许多领先的技术期刊和出版物上发表过很多主题的文章，从分层数据库技术和敏捷开发到虚拟化。

Agility Made Possible: CA Technologies 的优势

CA Technologies (NASDAQ: CA) 提供 IT 管理解决方案，帮助客户管理和保护复杂 IT 环境，以支持敏捷的业务服务。各组织可以利用 CA Technologies 软件和 SaaS 解决方案，从数据中心到云，加速创新、转换基础架构以及保护数据和身份。CA Technologies 承诺确保我们的客户通过使用我们的技术实现其预期成果和期望的业务价值。要详细了解我们的客户成功计划，请访问 ca.com/customer-success。有关 CA Technologies 的详细信息，请访问 ca.com。