42 | 实战：大型全球化电商的测试基础架构设计

2018-10-03 茹炳晟

讲述：茹炳晟
时长 15:42　大小 7.20M

你好，我是茹炳晟。今天我和你分享的主题是“实战：大型全球化电商的测试基础架构设计”。

在前面的两篇文章中，我和你分享了测试基础架构的设计以及演进之路，其中涉及到了统一测试执行平台、Selenium Grid 和 Jenkins 等一系列的概念。

在掌握了这些基础内容之后，今天我就和你一起看看大型全球化电商的测试基础架构又是如何设计的。这其中除了我之前介绍过的概念以外，还会引入一些新的服务和理念，我都会和你一一道来。

因为我们已经掌握了测试基础架构设计的基础知识，所以今天我会采用一种不同于以往由浅入深的方式，直接给出大型全球化电商网站的全局测试基础架构的最佳实践，然后再依次解

释各个模块的主要功能以及实现基本原理。

其实，**大型全球化电商网站全局测试基础架构的设计思路，可以总结为“测试服务化”**。也就是说，测试过程中需要用的任何功能都通过服务的形式提供，每类服务完成一类特定功能，这些服务可以采用最适合自己的技术栈，独立开发，独立部署。而至于到底需要哪些测试服务，则是在理解了测试基础架构的内涵后再高度抽象后得到的。从本质上来看，这种设计思想其实和微服务不谋而合。

根据在大型全球化电商网站工作的实际经验，我把一个理想中的测试基础架构概括为了一张图（如图 1 所示）。

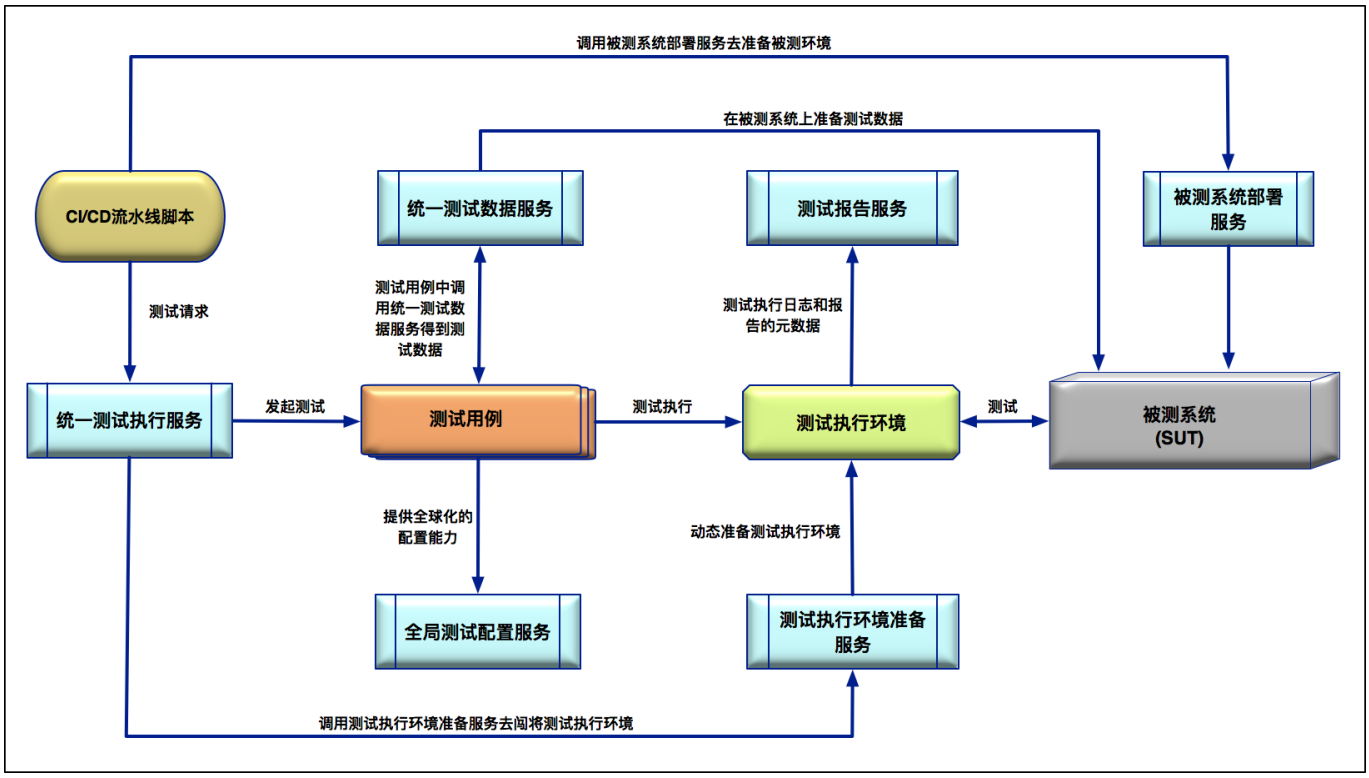

图 1 大型全球化电商网站的全局测试基础架构设计

这个理想的测试基础架构，包括了 6 种不同的测试服务，分别是：统一测试执行服务、统一测试数据服务、全局测试配置服务、测试报告服务、测试执行环境准备服务，以及被测系统部署服务。

接下来，我们一起看看这 6 大测试服务，具体是什么，以及如何实现。

统一测试执行服务

从本质上看，统一测试执行服务，其实和统一测试执行平台（你可以再回顾一下第 41 篇文章《从小工到专家：聊聊测试执行环境的架构设计（下）》）是一个概念。只不过，统一测试执行服务，强调的是服务，也就是强调测试执行的发起是通过 Restful API 调用完成的。

总结来说，以 Restful API 的形式对外提供测试执行服务的方式，兼具了测试版本管理、Jenkins 测试 Job 管理，以及测试执行结果管理的能力。

统一测试执行服务的主要原理是，通过 Spring Boot 框架提供 Restful API，内部实现是通过调度 Jenkins Job 具体发起测试。如果你对此还有疑惑，请参考第 40 篇文章《从小工到专家：聊聊测试执行环境的架构设计（上）》。

还记得我在前面一直提到的将测试发起与 CI/CD 流水线集成吗？这个统一测试执行服务采用的 Restful API 调用，主要用户就是 CI/CD 流水线脚本。我们可以在这些脚本中，通过统一的 Restful API 接口发起测试。

统一测试数据服务

统一测试数据服务，其实就是统一测试数据平台（你也可以再回顾一下第 37 篇《测试数据的“银弹”- 统一测试数据平台（上）》和第 38 篇《测试数据的“银弹”- 统一测试数据平台（下）》文章的内容）。

任何测试，但凡需要准备测试数据的，都可以通过 Restful API 调用统一测试数据服务，然后由它在被测系统中实际创建或者搜索符合要求的测试数据。而具体的测试数据创建或者搜索的细节，对于测试数据的使用者来说，是不需要知道的。也就是说，统一测试数据服务，会帮助我们隐藏测试数据准备的所有相关细节。

同时，在统一测试数据服务内部，通常会引入自己的内部数据库管理测试元数据，并提供诸如有效测试数据数量自动补全、测试数据质量监控等高级功能。

在实际工程项目中，测试数据的创建通常都是通过调用测试数据准备函数完成的。而这些函数内部，主要通过 API 和数据库操作相结合的方式，实际创建测试数据。

如果你对测试数据的准备还有疑问，或者想知道更多的细节内容，可以再回顾一下前面“测试数据准备”的系列的第 35~38 篇文章。

测试执行环境准备服务

测试执行环境准备服务，其实我也已经介绍过了。这里“测试执行环境”，是狭义的概念，特指具体执行测试的测试执行机器集群：对于 GUI 自动化测试来说，指的就是 Selenium Grid；对于 API 测试来说，指的就是实际发起 API 调用的测试执行机器集群。

测试执行环境准备服务的使用方式，一般有两种：

一种是，由统一测试执行服务根据测试负载情况，主动调用测试执行环境准备服务来完成测试执行机的准备，比如启动并挂载更多的 Node 到 Selenium Grid 中；

另一种是，测试执行环境准备服务不直接和统一测试执行服务打交道，而是由它自己根据测试负载来动态计算测试集群的规模，并完成测试执行集群的扩容与收缩。

被测系统部署服务

被测系统部署服务，主要被用来安装部署被测系统和软件。虽然这部分内容我以前没有提到过，但它很好理解。其实现原理是，调用 DevOps 团队的软件安装和部署脚本。

对于那些可以直接用命名行安装和部署的软件来说很简单，一般只需要把人工安装步骤的命名行组织成脚本文件，并加入必要的日志输出和错误处理即可。

对于那些通过图形界面安装的软件，一般需要找出静默（Silent）模式的安装方式，然后通过命令行安装。

如果被测软件安装包本身不支持静默安装模式，我强烈建议给发布工程师提需求，要求他加入对静默安装模式的支持。其实，一般的打包工具都能很方能地支持 Silent 安装模式，并不会增加额外的工作量。

被测系统部署服务，一般由 CI/CD 流水线脚本来调用。在没有被测系统部署服务之前，CI/CD 流水线脚本中一般会直接调用软件安装和部署脚本。而在引入了被测系统部署服务后，我们就可以在 CI/CD 流水线脚本中直接以 Restful API 的形式调用标准化的被测系统部署服务了。这样做的好处是，可以实现 CI/CD 流水线脚本和具体的安装部署脚本解耦。

测试报告服务

测试报告服务，也是测试基础架构的重要组成部分，其主要作用是为测试提供详细的报告。

测试报告服务的实现原理，和传统测试报告的区别较大。

传统的软件测试报告，通常直接由测试框架产生，比如 TestNG 执行完成后的测试报告，以及 HttpRunner 执行结束后的测试报告等等，也就是说测试报告和测试框架绑定在了一起。

对于大型电商网站而言，由于各个阶段都会有不同类型的测试，所以测试框架本身就具有多样性，因此对应的测试报告也是多种多样。而测试报告服务的设计初衷，就是希望可以统一管理这些格式各异、形式多样的测试报告，同时希望可以从这些测试报告中提炼出面向管理层的统计数据。

为此，测试报告服务的实现中引入了一个 NoSQL 数据库，用于存储结构各异的测试报告元数据。在实际项目中，我们会改造每个需要使用测试报告服务的测试框架，使其在完成测执行后将测试报告的元数据存入到测试报告服务的 NoSQL 数据库。这样，我们再需要访问测试报告的时候，就可以直接从测试报告服务中提取了。

同时，由于各种测试报告的元数据都存在了这个 NoSQL 数据库中，所以我们就可以开发一些用于分析统计的 SQL 脚本，帮助我们获得质量相关信息的统计数据。

测试报告服务的主要使用者是测试工程师和统一测试执行服务。对统一测试执行服务来说，它会调用测试报告服务获取测试报告，并将其与测试执行记录绑定，然后进行显示。而测试工程师则可以通过测试报告服务这个单一的入口，来获取想要的测试报告。

全局测试配置服务

全局测试配置服务是这 6 个服务中最难理解的部分，其本质是要解决测试配置和测试代码的耦合问题。这个概念有点抽象，我们一起看个实例吧。

大型全球化的电商网站在全球很多国家都有站点，这些站点的基本功能是相同的，只是某些小的功能点会有地域差异（比如，因当地法务、政策等不同而引起的差异；又比如，由货币符号、时间格式等导致的细微差异）。

假设，我们在测试过程中，需要设计一个 getCurrencyCode 函数来获取货币符号，那么这个函数中就势必会有很多 if-else 语句，以根据不同国家返回不同的货币符号。

比如，如图 2 所示的“Before”代码中，就有 4 个条件分支，如果当前国家是德国（isDESite）或者法国（isFRSite），那么货币符号就应该是“EUR”； 如果当前国家是英国（isUKSite），那么货币符号就应该是“GBP”；如果当前国家是美国（isUSSite）或者

是墨西哥（isMXSite），那么货币符号就应该是“USD”；如果当前国家不在上述的范围，那么就抛出异常。

图 2 全局测试配置服务的原理示例

上述函数的逻辑实现本身并没有问题，但是当你需要添加新的国家和新的货币符号时，就需要添加更多的 if-else 分支，当国家数量较多的时候，代码的分支也会很多。更糟糕的是，当添加新的国家时，你会发现有很多地方的代码都要加入分支处理，十分不方便。

那么，有什么好的办法，可以做到在添加新的国家支持时，不用改动代码吗？

其实，仔细想来，**之所以要处理这么多分支，无非是因为不同的国家需要不同的配置值**（这个实例中，不同国家需要的不同配置值就是货币符号），那如果我们可以把配置值从代码中抽离出去放到单独的配置文件中，然后代码通过读取配置文件的方式来动态获取配置值，这样就可以做到加入新的国家时，不用再修改代码本身，而只要加入一份新国家的配置文件就可以了。

为此，我们就有了如图 2 所示的“After”代码以及图中右上角的配置文件。“After”代码的实现逻辑是：通过 GlobalRegistry 并结合当前环境的国家信息来读取对应国家配置文件中的值。比如，GlobalEnvironment.getCountry() 的返回值是“US”，也就是说当前环境的国家是美国，那么 GlobalRegistry 就会去“US”的配置文件中读取配置值。

这样实现的好处是，假定某天我们需要增加日本的时候，getCurrencyCode 函数本身不用做任何修改，而只需要增加一个“日本”的配置文件即可。

至此，我们已经一起了解了大型全球化电商网站的全局测试基础架构设计，以及其中的 6 个主要测试服务的作用及其实现思路。现在，我再和你分享一个实例，看看这样的测试基础架构是如何工作的，帮助你进一步理解测试基础架构的本质。

大型全球化电商网站测试基础架构的使用实例

这个实例，我会以 CI/CD 作为整个流程的起点。因为，在实际工程项目中，自动化测试的发起与执行请求一般都是来自于 CI/CD 流水线脚本。

首先，CI/CD 流水线脚本会以异步或者同步的方式调用被测系统部署服务，安装部署被测软件的正确版本。这里，被测系统部署服务会访问对应软件安装包的存储位置，并将安装包下载到被测环境中，然后调用对应的部署脚本完成被测软件的安装。之后，CI/CD 脚本中会启动被测软件，并验证新安装的软件是否可以正常启动，如果这些都没问题的话，被测系统部署服务就完成了任务。

这里需要注意的是：

- 如果之前的 CI/CD 脚本是以同步方式调用的被测系统部署服务，那么只有当部署、启动和验证全部通过后，被测系统部署服务才会返回，然后 CI/CD 脚本才能继续执行；
- 如果之前的 CI/CD 脚本是以异步方式调用的被测系统部署服务，那么被测系统部署服务会立即返回，然后等部署、启动和验证全部通过后，才会以回调的形式通知 CI/CD 脚本。因此，CI/CD 脚本也要为此做特殊处理。

被测系统部署完成后，CI/CD 脚本就会调用统一测试执行服务。统一测试执行服务会根据之前部署的被测软件版本选择对应的测试用例版本，然后从代码仓库中下载测试用例的 Jar 包。

接下来，统一测试执行服务会将测试用例的数量、浏览器的要求，以及需要执行完成的时间作为参数，调用测试执行环境准备服务。

测试执行环境准备服务会根据传过来的参数，动态计算所需的 Node 类型和数量，然后根据计算结果动态加载更多的基于 Docker 的 Selenium Node 到测试执行集群中。此时，动态 Node 加载是基于轻量级的 Docker 技术实现的，所以 Node 的启动与挂载速度都非常快。

因此，统一测试执行服务通常以同步的方式调用测试执行环境准备服务。

测试执行环境准备好之后，统一测试执行服务就会通过 Jenkins Job 发起测试的执行。**测试用例执行过程中，会依赖统一测试数据服务来准备测试需要用到的数据，并通过全局测试配置服务获取测试相关的配置与参数。**

同时，**在测试执行结束后，还会自动将测试报告以及测试报告的元数据发送给测试报告服务进行统一管理。**

以上就是这套测试基础架构的执行过程了。

总结

通过前面几篇文章，我们已经掌握了测试基础架构的基础知识，所以今天我分享的主题就是，从实战的角度帮你夯实测试基础架构的基础。

其实，大型全球化电商网站全局测试基础架构的设计思路，可以总结为“测试服务化”。于是，我总结了一个比较理想的测试基础架构，应该包括 6 大服务：统一测试执行服务、统一测试数据服务、全局测试配置服务、测试报告服务、测试执行环境准备服务，以及被测系统部署服务。

其中，统一测试执行服务，本质上讲就是统一测试执行平台；统一测试数据服务，其实就是统一测试数据平台；测试执行环境准备服务，指的是狭义的测试执行环境准备。这几部分内容，我都已经在前面的文章中分享过了，如果你有任何问题，也可以再给我留言一起讨论。

而被测系统部署服务，主要是被用来安装部署被测系统和软件，这部分也很简单；测试报告服务，虽然和传统的测试报告区别较大，但也可以通过引入一个 NoSQL 数据库，以存储的测试报告元数据的方式去实现。

全局测试配置服务是这 6 个服务中最难理解的部分，其本质是要解决测试配置和测试代码的耦合问题。我通过一个具体的不同国家对应不同货币符号的例子，和你讲述了具体如何解耦。

思考题

除了我今天分享的 6 大服务以外，其实还有更多的服务可以帮助我们提升测试效能，比如全局 Mock 服务、工程效能工具链仓库等等。你还能想到有哪些与测试相关的服务吗？

感谢你的收听，欢迎你给我留言一起讨论。

上一篇　41 | 从小工到专家：聊聊测试执行环境的架构设计（下）

下一篇　43 | 发挥人的潜能：探索式测试

精选留言 (8)

写留言

颜瑞
2018-10-11

2

一直疑惑于测试用例的编写方式，按照您的说法，用例是jar包形式管理，那用例是用junit写的吗？还是单独写了套框架？

作者回复: 在这个例子中，测试用例在测试框架的支持下是被打包成jar文件的。

Bob_jc
2019-05-30

测试报告服务器，要简单统一，可以看看开源的allure report 。很多测试框架都有对应的，即使没有也可以自己去实现测试结果写成框架支持的json.我现在倾向于自己写一个类似于测试报告（allure report 的页面展现、布局交互和美观都很优秀）。

口水窝
2019-05-16

感觉今天的课程就是把以前的被测系统部署、测试数据准备、执行机器、全球配置、执行测试、测试报告整个流程串讲了一下，视野宽阔了好多，后面老师说的全全局Mock服务、工程效能工具链仓库都没有听说过，捂脸!

展开

小葱拌豆腐
2019-03-08

我以前给我们的项目定位是中型项目；但是现在我改主意了。虽然有点受打击，但是开了眼界，脚踏实地，路还很长。

arthur
2018-12-09

我们的统一测试部署服务，是一个叫做instanceup的系统，只要配置一下你想安装的环境地址和数据库地址，点一下按钮就完全能自动部署我们的软件

小老鼠
2018-11-29

Python 有无案例

展开

作者回复: python没有案例

小老鼠

2018-11-29

每个大型电商自动化测试平台区别大吗?

展开

作者回复: 每家的做法都不一样，取决于所处的阶段

胖虫子

2018-11-16

这个只能膜拜了，要实现的多少成本

展开