

Tableau Server 可扩展性

面向服务器管理员的技术指南

Neelesh Kamkolkar
产品经理，数据和性能部门

目录

执行摘要.....	3
VizQL 颠覆陈旧模式	5
Tableau 架构	7
统一内存与实时技术的架构.....	8
测试方法和方法体系.....	9
方法体系.....	9
真实工作负载特征描述	11
测试建模步骤	11
后台程序方法体系.....	12
标准化的隔离环境.....	13
部署拓扑.....	14
测量与报告.....	15
方案.....	15
响应时间.....	16
方案吞吐量.....	16
活动用户.....	17
结果	18
Tableau Server 10 可实现线性扩展	18
后台程序结果	22
隔离后台程序进程.....	26
后台程序方面的注意事项	28
最佳做法 – 自行进行扩展测试.....	28
实际环境优化最佳做法	29
总结	30

执行概要

对许多组织而言，Tableau 发挥着举足轻重的作用。这些组织中的许多部门通过从数据中发现见解来创造巨大价值。因此，IT 团队与业务部门通力合作，部署 Tableau 作为整个企业的分析平台。当企业开始以这种方式部署 Tableau 时，企业的架构师和 IT 领导必须理解 Tableau Server 如何随数据、内容和用户的增加而进行扩展，如何跨各种异构的企业 IT 平台进行部署和集成，从而为企业当前和将来的分析需求提供支持。

本白皮书面向企业架构师和 IT 技术领导。它深入地介绍了 Tableau 的架构，以及 Tableau 如何随工作负载的不断增长进行扩展。

我们探索并测试了 Tableau Server 10 的扩展功能，并将此次的测试结果与该产品的早期版本进行对比。根据大家的要求，我们扩大了可扩展性测试的范围，使其不但包括用户可扩展性，还包括后台工作负载。

Tableau Server 的性能和可扩展性受一系列系统因素的影响。一些重要的系统变量包括工作簿设计、服务器配置、基础架构调整、数据环境、计算容量和网络。在不同的用户配置和部署中，这些因素差异很大。任何可扩展性测试的结果都会随着这些变量的调整而变化而有所差异。为了降低这些变量的影响并对其进行隔离，我们在物理计算机上的封闭网络实验室中进行测试。我们的目标是将外部系统的影响造成的测量差异降至最低。随后我们就可以对 Tableau Server 的实际使用情况进行建模，以此测量各种可扩展性指标。

为此，我们首先对处于峰值使用时段的 Tableau Server 真实生产部署进行分析，然后通过自动测试对使用情况进行建模。这种两步式方法模拟了非常真实的工作负载。这种方法还模拟了实际用户和后台程序工作负载（主要是数据提取和用户通知）对系统造成的实际影响差异。这里的差异包括用户与可视化进行的各次交互之间的等待时间，以及按照计划运行的后台程序作业的数量。

我们在测试中对这些差异进行了建模，以此模仿真实的生产条件。我们发现，当我们在 Tableau Server 群集中添加更多工作节点时，系统实现了线性扩展。我们的实验对服务器施加了峰值负载条件，这超出我们在生产环境中观察到的负载。在生产环境中，使用量往往会在一个工作日中呈现出多个频率高、持续时间短的峰值爆发期。我们按吞吐量来测量负载。吞吐量是服务器在一个给定时段内处理的工作量，简单地说，就是每秒处理的事务数。如下图所示，我们在实验中观察到，吞吐量从每秒处理 4 个事务扩展到每秒处理 18 个事务。每个条形表示一个实验，列标题说明运行这些实验时采用的拓扑（独立服务器、主服务器 + 1 个工作服务器等）。

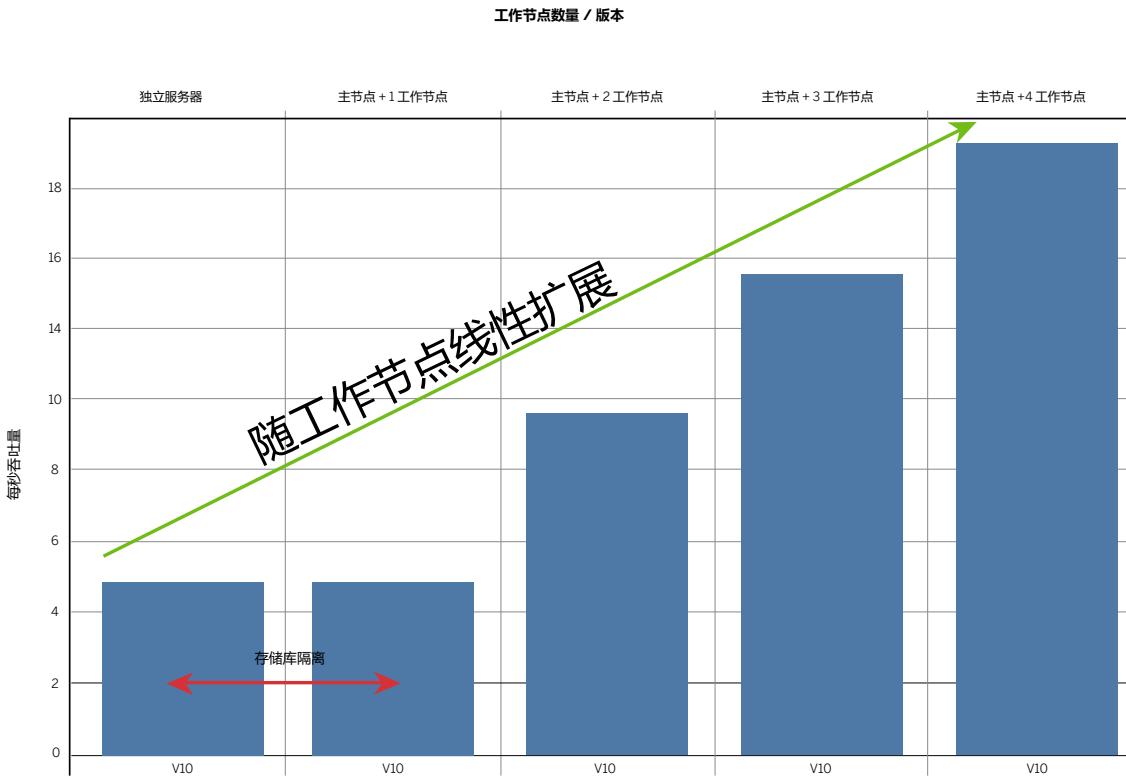


图 1：每秒方案吞吐量

我们确定了具有 8 个物理核心的单个服务器能够处理的最大活动用户负载数。活动用户负载指的是针对 Tableau Server 的一组自动化操作。本白皮书将在后面部分详细说明一个活动用户负载包括哪些操作。测量具有 8 个物理核心的单个服务器能够支持的最大活动用户数之后，我们使用此活动用户数作为扩展单位来进行测试，确定我们能否通过向群集添加同构 8 核工作节点来实现负载的线性增加。

最终我们观察到，在根据实际情况建模的持续峰值负载下，单个 8 核服务器最多可以支持 112 个活动用户。当我们将基线进行线性扩展 — 添加各自包含 8 个核心的 4 个同构工作节点以及 1 个只包含存储库和基本安装的主控制器节点后，我们能够支持 448 个活动用户。这些用户同时在系统中工作。因此，如果您考虑到所有用户并不总是会同时使用系统并以此进行推算，那么具有 8-32 个核心的群集配置预计可以支持 500 至 10,000 名用户。我们将看到，这种推算取决于多个不同参数。所有其他变量相同时，支持特定数量的用户所需的服务器大小取决于分析使用方式和数据条件。我们的测试表明，Tableau 架构让您可以通过在 Tableau Server 群集中添加更多工作节点来不断实现用户群的线性扩展。

在我们的测试中，我们不断增加负载，确定服务器从何时开始返回错误或超时。这项压力测试让我们可以定义服务器可扩展性和性能的上限。在对这些限制进行定量后，我们不断对设计施压，使之超过典型的生产爆发负载，并维持在峰值负载水平。在实际操作中，负载通常低于峰值且状态较为稳定，因此，平台可以支持的使用量应该大于本文提出的值。您可以确信，由于 Tableau Server 架构可以线性扩展，因此它能够通过扩展满足您当前的需求并适应您未来的增长。

下表对每种拓扑和用例进行了说明。“风险特征”列对应一系列变量，例如根据自动故障转移、硬件故障风险以及峰值爆发储备空间等不同因素，该拓扑面临的风险是会增大还是减小。

部署配置	用例	风险特征
独立服务器	简单服务器部署	可用性风险高，资源竞争程度高，无峰值储备空间 负载性能挑战
1主节点 + 1工作节点	二节点部署	可用性风险高，与存储库的资源竞争程度较低， 无峰值使用储备空间
1主节点 + 2工作节点	三节点部署	可用性风险中等，资源竞争程度较低， 水平扩展性能改善， 峰值使用储备空间很小
1主节点 + 3工作节点	四节点部署	可用性风险中等，资源竞争程度较低， 扩展性能改善，中等峰值使用 储备空间
1主节点 + 4工作节点	五节点部署	可用性风险较低，扩展性能改善， 峰值使用储备空间可用 ¹

表 1：部署拓扑和用例

¹可用峰值储备取决于许多因素并且无法保证

必须结合具体方法和测试来理解这些数字。为此，本白皮书剩余部分将重点介绍架构、Tableau 架构与传统商业智能技术相比有何独特之处，以及我们的方法和可扩展性结果分析。

VizQL 颠覆陈旧模式

如果您习惯于传统商业智能 (BI) 解决方案或者刚开始接触 Tableau，则了解 Tableau 在工作原理方面与传统商业智能之间的一些核心差异或许会对您有所助益。传统商业智能的方法是“先进行查询、再进行可视化”，我们已经对这种方法的基础假设提出了挑战。我们相信，对数据的实时探索和提问才是揭示见解的正确途径。十多年来，Tableau 一直包含一项称为 VizQL™ 的专利技术，该技术可以将查询和可视化合并到同一个平台中。这对经过组合的强大功能让最终用户可以在进行数据可视化的同时，不受限制地提出问题 - 查询、筛选和分析。VizQL™ 是 Tableau 的基础语言，它可以表达用户的问题和操作，将其转化为查询，这些查询可以针对企业内部或云端的任何数据集运行。持续十多年的工程投资让这项技术日臻成熟，并且它还在不断发展，为新一代的数据源和分析要求提供支持。

与根据一组预定义的静态需求而设计和开发的传统商业智能报告不同，Tableau 可视化专为交互和协作而打造。用户无需编写复杂的 SQL 查询或联接即可提出任何数据问题。而且，用户可以很快获得问题的答案，而无需等待传统软件开发流程开展到下一个周期。他们能够对现有的可视化进行迭代，然后继续进行分析，回答关于自己业务和项目的问题。

使用传统商业智能时，您可能习惯于对符合特定服务级别协议的静态报告进行负载测试，其中的查询是针对具有预构建优化项、面向特定目标的系统而设计的。静态报告具有固定的范围和一组固定的查询，往往由开发人员花数周时间逐一优化。虽然为静态报告定义和设置 SLA 相对比较轻松，但报告的更改会重新启动整个开发流程，这是一个通常未在 SLA 中加以考虑的限制。

而 Tableau 可视化则可以根据用户执行的试探性操作，重新生成和提交新的查询。VizQL 中的本地浏览器缓存和优化可以实现快速数据检索，用户无需因为等待查询结果而中断分析流程。

VizQL 已经作为一种语言包含到我们的 VizQL 服务器进程中，并且能够与许多其他分布式服务器进程协同工作，提供可扩展、可用、安全和可靠的商业分析平台。

Tableau 架构

Tableau 具有灵活的架构，可实现纵向和横向扩展。因此，Tableau 可以作为企业标准平台运行，也可以用作云分析支持平台。Tableau Server 能够支持极为复杂的企业生产架构要求，也可以支持简单的部门或工作组级别部署。

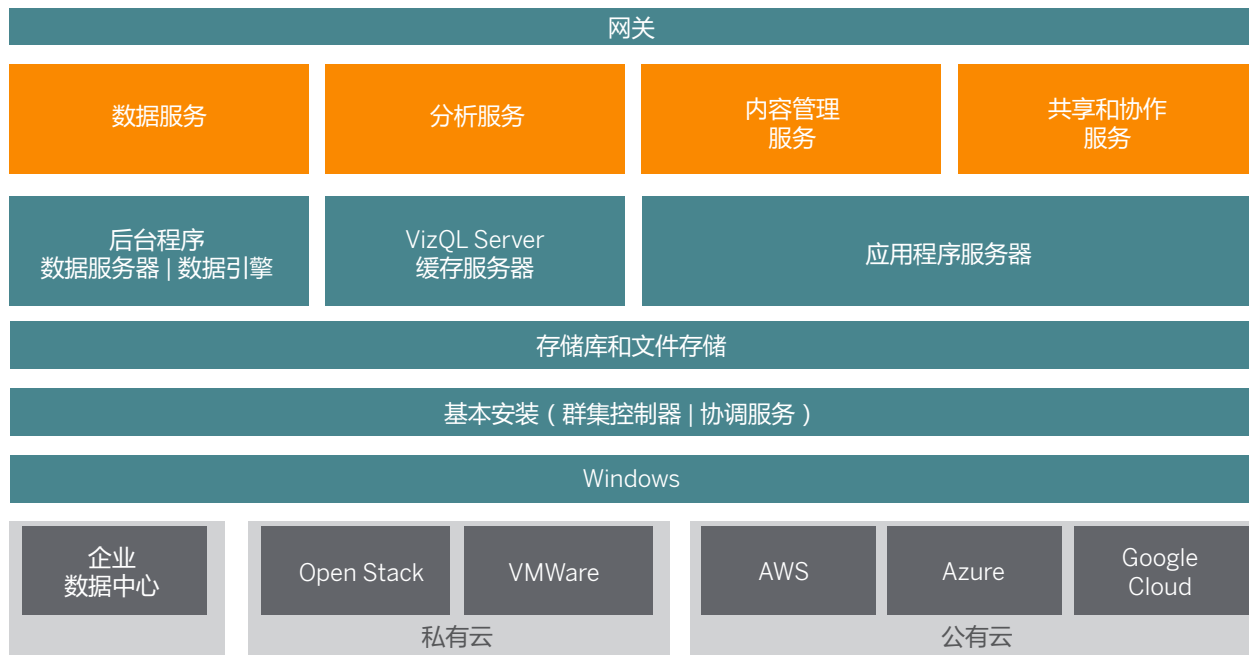


图 2：Tableau Server 架构

Tableau Server 具有简单的安装和配置流程。安装完成后，多个服务器进程（图 2 中以蓝色显示）相互协作，在不同层级提供服务。

网关进程是其中一个组件，它将来自所有 Tableau 客户端的流量重新定向到群集中可用的服务器节点。

数据服务是服务的逻辑分组，这些服务可以提供最新数据、共享元数据管理、管控数据源和内存数据。驱动数据服务的基础进程为后台程序进程、数据服务器进程和数据引擎进程。

分析服务由 VizQL 和缓存服务器进程组成，提供面向客户的可视化、分析服务和缓存服务。

内容管理以及共享和协作服务由应用程序服务器进程驱动。用户登录、内容管理（项目、站点、安全许可等）之类的 Tableau Server 核心功能和管理活动由应用程序服务器进程提供。

上述所有服务都使用并依赖于存储库进程，该进程包含结构化关系数据，例如权限、工作簿、数据提取、用户信息和元数据等。文件存储进程可以在整个群集上实现数据提取文件冗余，并确保数据提取可以在所有群集节点上进行本地使用。当负载较大时，数据提取文件可以在整个群集上进行本地使用，实现更快的处理和呈现。

Tableau 的灵活架构使您可以在任何位置运行平台。您可以将 Tableau Server 安装在本地，私有云或数据中心，或者 Amazon EC2™、Google Cloud Platform™ 或 Microsoft Azure™ 上。Tableau 分析平台还可以在 VMware ESXi™ 或 Microsoft Hyper-V™ 之类的虚拟化平台上运行。我们建议您遵循每个虚拟化平台的最佳做法，确保 Tableau Server 实现最佳性能。

有关各个服务器进程的详细信息，请参阅 Tableau Server [管理指南](#)。

统一内存与实时技术的架构

Tableau 提供真正的异构平台支持，可以连接来自几十家供应商的 50 多种最流行的数据源。该架构非常灵活，您可以选择进行速度极快的内存中分析，也可以直接连接到实时数据存储来进行分析。不管选择哪种途径，您都可以根据不同的业务需求和数据源在实时连接和内存中分析之间轻松切换。

在为内存中分析提供支持时，Tableau 会将数据提取到名为 Tableau 数据提取 (TDE) 的专有列存储中，而列存储将加载到内存映射文件中，以便进行快速访问。数据提取可由用户在 Tableau Desktop 上创建，也可以使用 Tableau API 从外部业务流程创建。Tableau Server 架构提供内置支持，确保用户创建的数据提取按照受您控制的、经过优化的计划进行更新并保持常新。

无论如何连接到数据并将其导入 Tableau Server，您必须确保自己有足够的内存资源来进行数据分析、缓存、数据提取刷新以及其他相关操作。

与单纯的内存中工具不同，Tableau 的内存占用量分散在整个群集中，而不是集中在单个服务器上。内存的使用通过缓存服务器（提供外部共享查询缓存和存储会话级别缓存的其他进程）分布在整个群集中。因此，负载增加所带来的内存影响将根据您的工作负载分散在整个群集中。

测试方法和方法体系

在多数传统负载测试项目中，目标应用程序被视为一个黑盒。与此不同，对 Tableau Server 进行负载测试时，必须对其架构有足够的理解。Tableau Server 的构建方式使其可以在任何位置运行：本地或云端。它可以为任何规模的团队和组织提供服务。因此，对于大多数标准部署，Tableau Server 的默认安装应该能够起到很好的效果。在尝试扩展 Tableau Server 并将其部署到整个组织中时，您必须理解不同工作负载的处理方式，以及如何通过一些简单的配置调整来改善部署方案的效果。

鉴于版本 10 的可扩展性目标，我们准备回答以下问题：

1. 通过为以下两种常见方案添加更多硬件，Tableau Server 10 能否实现线性扩展：

- 扩展最终用户工作负载
- 扩展后台程序工作负载

2. 何时将后台程序进程移至专用的工作节点比较好？

3. 与此前的版本相比，Tableau Server 10 的性能如何？

本白皮书此前的版本特别关注最终用户可扩展性，但许多客户对我们提出了如何扩展后台程序工作负载的问题。后台程序工作负载控制数据的新鲜度（数据提取刷新）以及分析的使用范围（订阅通知）。

方法体系

版本 10 的 Tableau Server 以提高性能和可扩展性为主要目标。因此，制定一种企业级的生产就绪型测试方法体系也是一个核心要求。从 v10 的预发布版本开始，我们就已在采用本部分说明的方法体系。在敏捷开发的迭代流程中，我们发现并修复了这种方法体系揭示的二十多个扩展和性能方面的漏洞。我们在版本 v10 发布后继续进行测试，并在 10.0.1 维护版 (MR) 中修复了诸多漏洞。本白皮书基本上通篇在谈 v10 版本，但测试结果基于 10.0.1 MR。要最大程度发挥 v10 在扩展和性能方面的优势，您应该在组织中运行 10.0.1 MR 或更高版本。

我们不断改进自己的可扩展性做法，以便收集和测试能够代表真实客户情景的工作负载。许多变量都会影响部署的可扩展性，但在进行部署规划时，应该考虑的重要因素有：

- 用户影响 - 自助服务的使用情况和用户采用率：将有多少用户使用分析技术？用户将多久使用一次分析技术来作出明智决策？用户所创建的可视化的复杂程度如何？
- 数据影响 - 新鲜度、规模和位置：您的数据规模有多大？这些数据位于何处？这些数据需要达到何种新鲜程度才能为业务决策提供准确的依据？

有效业务决策所需的分析使用量	高 (每秒)	7.示例： WW 数据探索 Tableau Public (美国总统大选) 30K 次查看/小时	8.示例： 销售配额仪表盘， 电视上的 Tableau	9.示例： 空中交通管制员， 监控财务， 贸易执行
	中 (每小时一次)	4.示例： 每日商店库存 保险客户 分析营销 (目标)	5.示例： 患者容量 经销商管理	6.示例： 支持升级 仪表盘，财务 组合仪表盘 欺诈调查
	低 (每天一次)	1.示例： 工程 - 船房抵押贷款 库存 传统商业智能	2.示例： 谁最诱人 销售线索跟踪	3.示例： 公路 Web 交通仪表盘
		低 (每天一次)	中 (每小时一次)	高 (每秒)
有效业务决策所需的数据刷新频率				

图 3：分析使用频率和数据刷新频率矩阵

为了同时测试这两个考虑因素，我们需要进行测试整合。在这方面，我们逐渐增加服务器的最终用户数量，并增加后台程序工作负载。通过这种方法，我们可以研究后台程序工作负载对最终用户服务质量的影响。为了对低度、中度和高度分析使用频率以及隔离后台程序工作负载造成的影响进行建模，我们在自己的实验室中采用各种工作负载对测试进行了 400 多次迭代运行。我们研究了系统可扩展性，同时修复了一些只在高负载下才表现出来的错误。

为了让测试能够反映真实情况，我们首先需要采集一组合适的工作负载来运行这些测试。

真实工作负载特征描述

针对 v10，我们观察了真实生产环境的 Tableau Server 在峰值使用期的使用情况并进行了特征描述。

为了确定用于建模的工作负载以及测试的工作负载特征，我们分析了若干 Tableau Server 日志文件，这些文件来自一个包含 3000 名用户的生产环境。我们找出了使用量极大的可视化和工作簿。我们计算了使用率在这些工作簿间的分布情况，然后分析了使用率特征。例如，我们调查了请求之间的时间间隔（即思考时间）。在对生产服务器的工作负载进行建模时，我们采用了以下具体步骤。

测试建模步骤

1. 获取峰值使用时段的生产服务器日志
2. 确定在服务器上占用的加权平均时间最多的前 N 个工作簿视图。
3. 加权平均值 = 平均响应时间 * 请求数
4. 计算前 N 个工作簿视图的相对权重
5. 针对选择的每个工作簿视图，
 - a. 找出满足以下条件的可视化工作负载的百分比：刷新 = y（这种方法可以确定，有多少名用户正在主动进行数据刷新以获取最新数据）
 - b. 按权重找出前 N 次交互
 - c. 计算交互之间的平均思考时间
6. 为了进行模型验证：
 - a. 确定工作簿视图的平均引导间时间 TBB（测试间时间）
 - b. 确定在服务器上占用的加权平均时间最多的前 N 个后台程序任务
7. 按不同的任务对后台程序流量进行归类（例如订阅和数据提取刷新）
 - a. 确定前 N 个后台程序任务间的平均时间并将其包含在模型中
 - b. 确定不同计划的订阅数量
 - c. 确定数据提取刷新的规模和类型（发布到数据服务器的数据提取、工作簿数据提取）

我们使用上述所有数据对真实的混合工作负载建模，该模型可以反映真实用户在峰值使用期间如何使用生产服务器。最后，我们根据生产日志分析，为后台程序数据提取刷新和订阅生成了基于工作负载的模型。

下表总结了这组混合的工作负载。

工作负载名称	说明	如何与 此前版本进行对比
真实生产 Server 工作负载	此工作负载基于对生产 Tableau Server (为一个组织中的 3000 名用户提供服务, 是 IT 管理的任务关键型应用程序) 使用情况的分析和特征研究。	仅能按照此处的说明在 9.3 和 10.0 之间对比此白皮书中的结果数据。这些结果不能与我们此前发布的白皮书进行对比—包括 9.0 白皮书—因为测试方法体系有显著区别。
真实生产 Server 工作负载 + 新功能	此工作负载将上述工作负载与运行 v10 中新功能的工作簿合并。	不能在 9.3 和 10.0 之间进行对比, 因为 9.3 无法运行 v10 功能。本白皮书中结果的最佳用途是提供关于 v10 扩展特点的信息。
Backgrounder 混合组	此工作负载基于生产工作负载分析, 以真实工作簿和计划为建模对象, 反映生产环境。	10.0 可扩展性测试白皮书中引入的新混合组与此前任何白皮书的结果无可比性。

表 2：工作负载组合描述

随后, 我们在位于物理计算机上的一个隔离的可扩展性实验室中单独运行每个组合, 逐渐增加用户数量和后台程序负载。群集达到满负载状态后, 我们会每次添加一个工作节点, 然后继续扩展负载, 并对系统在每个实验期间的表现进行观察。我们在每次运行期间记录关键性能指标, 例如响应时间、吞吐量和错误率。我们还使用 JMX 记录系统指标和应用程序服务器指标。对于每次运行, 我们都寻找数据的相关性, 并分析系统在工作负载不断增加时的表现。同时, 我们还在敏捷开发过程中发现并修复了一系列可扩展性错误。

后台程序方法体系

后台程序服务器处理系统级别和用户级别的后台作业。例行存储库维护任务之类的系统级别作业由后台程序执行。用户级别作业是用户可能提交给系统, 让系统代表该用户运行的作业。例如, 用户可以将数据提取发布到服务器, 然后为该数据提取配置根据计划定期发生的数据刷新操作。这组操作将创建一个刷新作业。后台程序是一个查看作业列表并代表用户执行作业的进程。这种方案对于自助功能的有效实施至关重要, 因为用户无需等待管理部门刷新数据。然而, 如果管理 Tableau Server 的管理团队不对此类负载进行容量规划, 最终用户可能会因此遇到服务质量问题。在进行服务器大小调整和规划时, 一个关键环节是确定您需要在多大程度上优化后台程序服务。您应该考虑是否要在另一台计算机上单独运行后台服务, 以便隔离这些工作负载。

本白皮书在末尾部分介绍了一些简单的最佳做法，它们可以帮助您根据一天中的时段采取不同的工作负载处理方式。但是，如果您在同一台计算机上运行后台程序和分析服务，您可能会发现最终用户服务质量在高负载情况下受到影响，这是服务器上的资源共享限制造成的。

因此，我们想要研究当后台程序与分析服务在同一台计算机上运行时，它将对最终用户扩展产生怎样的影响。此外，我们还想以定量方式调查隔离到专用硬件上的后台程序如何随负载的增加而扩展。

为了对此进行研究，我们使用具有 4 个物理核心的计算机单独运行后台程序。我们没有在同一台计算机上运行任何其他 Tableau Server 进程。我们在版本 9.3 和 10.0 的 Tableau Server 上运行相同的、根据生产环境建模的工作负载。此工作负载包括数据提取刷新和订阅，这样我们就可以重点关注用户级别作业。此工作负载包括 8 个计划的 400 个订阅。我们研究了订阅成功通知以及 Tableau 完成所有订阅需要的时间。

标准化的隔离环境

我们在性能实验室中，采用完全相同的、具有以下规格的物理机运行了这些可扩展性测试。

服务器类型	Dell PowerEdge R620
操作系统	Microsoft Windows Server 2012 R2 标准 64 位
CPU	2.6 GHz 1x8 物理核心，启用超线程（16 逻辑核心）
内存	64 GB

表 3：测试环境中每个节点的硬件规格

虽然表中列出了物理核心，我们建议您不要禁用超线程。为了保持一致性，除此处的说法外，本文其余部分将使用物理核心计数的说法并总是假定物理机上已经启用超线程。

部署拓扑

群集由一个或多个主（控制器）节点和一个或多个工作节点组成。在我们的测试中，工作节点具有相同的进程配置：

进程	主节点 tsperf-212.perf.dev.tsi.lan	工作节点 1 tsperf-213.perf.dev.tsi.lan	工作节点 2 tsperf-215.perf.dev.tsi.lan	工作节点 3 tsperf-216.perf.dev.tsi.lan	工作节点 4 tsperf-217.perf.dev.tsi.lan
群集控制器	✓	✓	✓	✓	✓
网关	✓	✓	✓	✓	✓
应用程序服务器		✓	✓	✓	✓
VizQL Server		✓✓	✓✓	✓✓	✓✓
缓存服务器		✓✓	✓✓	✓✓	✓✓
搜索和浏览		✓	✓	✓	✓
后台程序		✓	✓	✓	✓
数据服务器		✓✓	✓✓	✓✓	✓✓
数据引擎		✓	✓	✓	✓
文件存储		🔄 正在同步	🔄 正在同步	🔄 正在同步	🔄 正在同步
存储库	✓				

主动
 忙
 被动
 未许可
 停机
 状态不可用

图 4：进程配置

此处显示的工作进程配置为默认配置。您可能会获得更高或更低的可扩展性，具体取决于您为自己的环境和使用情景配置的进程的数量和类型。

主节点配置有基本安装，这包括群集控制器、网关和存储库进程。值得注意的是，如果按核心许可方案进行部署，核心计数不包括这种类型的主节点配置。

我们使用负载生成器 (TabJolt) 扩展工作负载，以此来模拟前文所述的用户工作负载。TabJolt 是一款“即点即运行”的负载和性能测试工具，专门用于 Tableau Server 9.0 或更高版本。下图显示的是测试执行的逻辑视图。

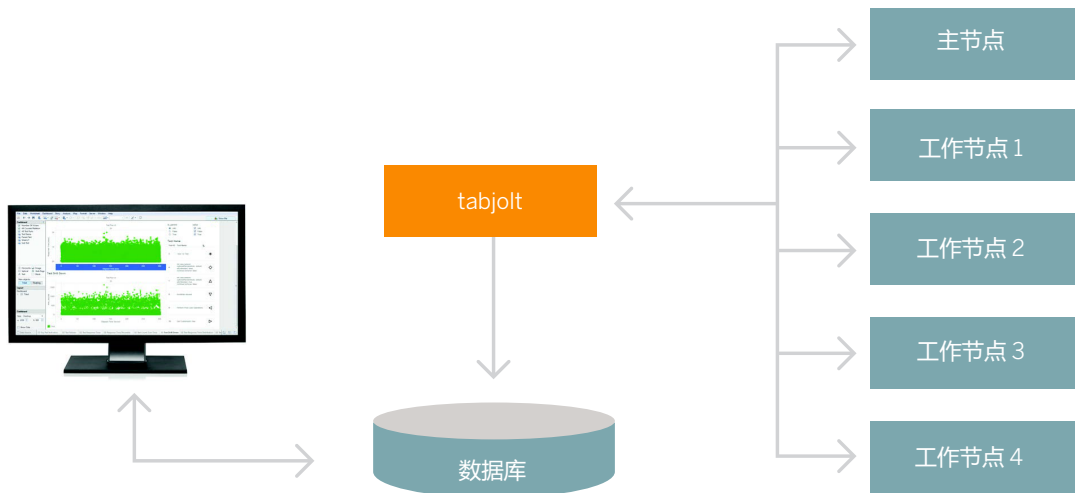


图 5：测试环境逻辑视图

我们从每个测试迭代收集了数据进行分析。但在开始讨论结果之前，我们先要了解其中一些指标和定义。

测量与报告

为了解硬件性能和可扩展性，我们测量了多个指标，包括 CPU、内存和磁盘的系统指标。我们还测量了响应时间、吞吐量、错误率、运行持续时间之类的性能和可扩展性指标。

让我们快速介绍一些定义，以便了解本白皮书中讨论的数据。

方案

方案是服务器上最高级别的用户活动。在此前的白皮书版本中，我们重点关注启用来宾访问的服务器上的可视化负载和交互时间。在当前的白皮书版本中，我们扩大了工作负载范围，使其包括应用程序服务器服务（例如登录）和其他服务。最终用户根据生产工作负载建模执行一系列步骤，我们使用自定义版的 TabJolt 对此进行了模拟。

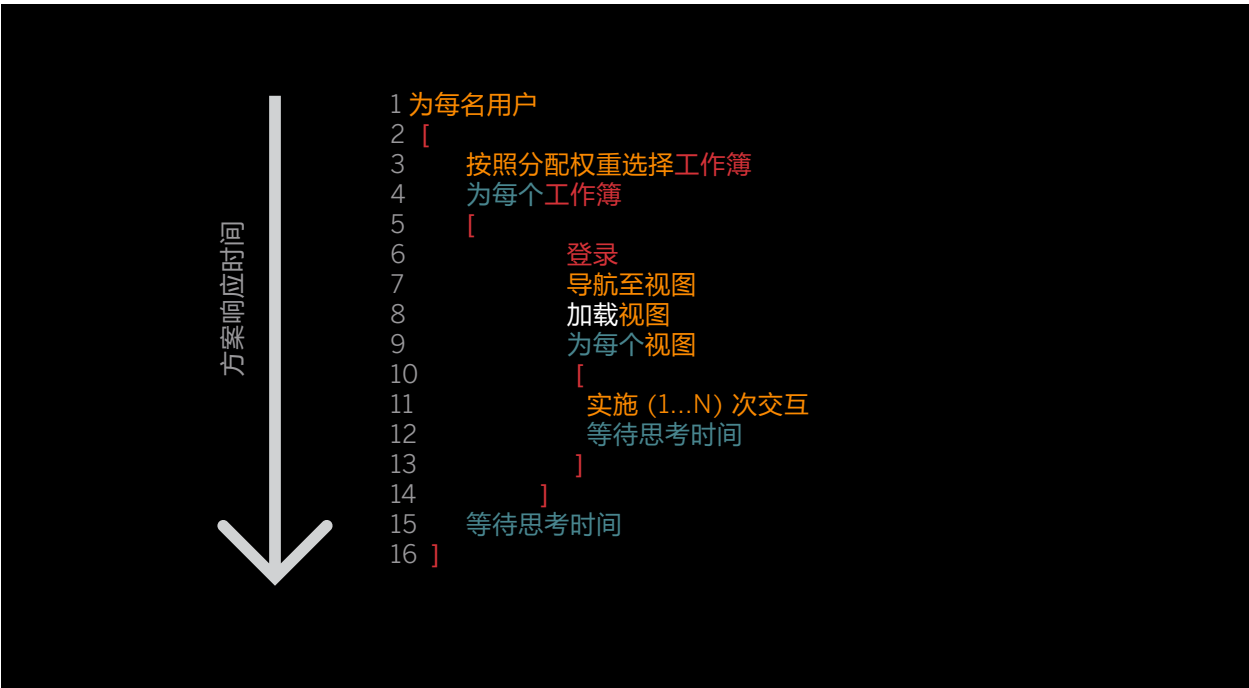


图 6：方案测试算法

本白皮书中报告的响应时间超过了较简单的负载和交互模型的响应时间，因为我们的测试模型包含完成方案所需的时间。我们从客户的角度来测量和报告事务的响应时间。这意味着，方案会从真实最终用户的视角进行描述，包括延迟等环境网络变量。

响应时间

响应时间是服务器对最终用户请求作出响应所需的秒数。请考虑以下示例：一名用户登录到服务器，导航至可视化，更改了该可视化的筛选器，等待该可视化进行更新和呈现，然后分析该可视化（思考时间）。我们会测量从该用户登录服务器到用户思考时间结束时这一总体“时钟时间”，并将其报告为此次方案迭代的响应时间。

方案吞吐量

方案吞吐量是每秒成功完成的方案数量 (TPS)。我们通过如下方法计算方案 TPS：采用特定拓扑，以系统的计算上限运行方案一小时。方案总数除以一小时内的秒数 (3,600) 即可得到 TPS。

例如，在测试运行时间内，我们在一个 8 核 Tableau Server 10 上完成了 16,372 个方案。这可以计算出 TPS 约为 4.5 (16,372/3,600)。这些实验将服务器的负载推高，使其远远超出我们在生产环境中观察到的负载水平（其中 TPS < 1）。实验数据之所以显著增大，是因为我们将服务器推至极限，以便了解所选硬件和扩展单元的终极性能容量。

然而，生产部署在峰值使用期会出现负载爆发，用户的操作速度通常要远低于自动测试执行框架。请记住这种差异。例如 Tableau Public 可以提供每秒大约 12 个可视化负载（称为“展示次数”）的超大规模，相当于每周展示超过 700 万个 Tableau 可视化。

活动用户

指标“活动用户数量”用于测量在一小时的峰值时间窗口内，同时使用 Tableau Server 的用户的数量。现在，方案包括登录、可视化负载、用户交互、搜索和其他操作。我们将活动用户定义为在一小时的峰值时间窗口内执行上述任何一种操作的用户。

为了确定我们的 Server 安装可以支持多少名活动用户，我们首先确定了单个 Server 在不引起响应时间恶化、不导致错误率（Tableau Server HTTP 错误）超过 2% 或 CPU 使用率超过 80% 的情况下能够支持多少名用户。

我们根据自己的生产工作负载特征描述，为特定工作簿分配权重。随后，我们生成一个线程（虚拟用户），该线程将根据分配的权重随机选择工作簿，然后完成前文所述的整个方案。方案结束时，该线程将等待指定的思考时间。这段时间结束后，该线程将完成其迭代并退出。我们在测试过程中测量了方案吞吐量、响应时间、错误率、CPU 和内存使用率以及其他一些指标。只要 CPU 使用率仍然低于 80%，错误率低于 2% 且响应时间没有恶化，我们就不断增加活动线程数。上述任何一个指标达到阈值时，我们将其视为该拓扑可以合理支持的最大活动用户数量。

为了验证 Server 能否线性扩展，我们先确定单个计算机的最大活动用户数量，然后按增量逐步增加虚拟用户的数量，并验证我们预设的条件是否依然能在新负载下得到满足。

结果

至此我们已经了解了执行测试的方式、所使用的部署以及各项指标，现在就来了解结果如何。

Tableau Server 10 可实现线性扩展

我们的第一个问题是 Tableau Server 10 如何扩展？我们观察到，通过在群集中添加更多的工作节点，Tableau Server 10 可随用户负载的增加实现线性扩展。下图显示了每秒完成的用户方案数量在工作节点增加时的变化情况。

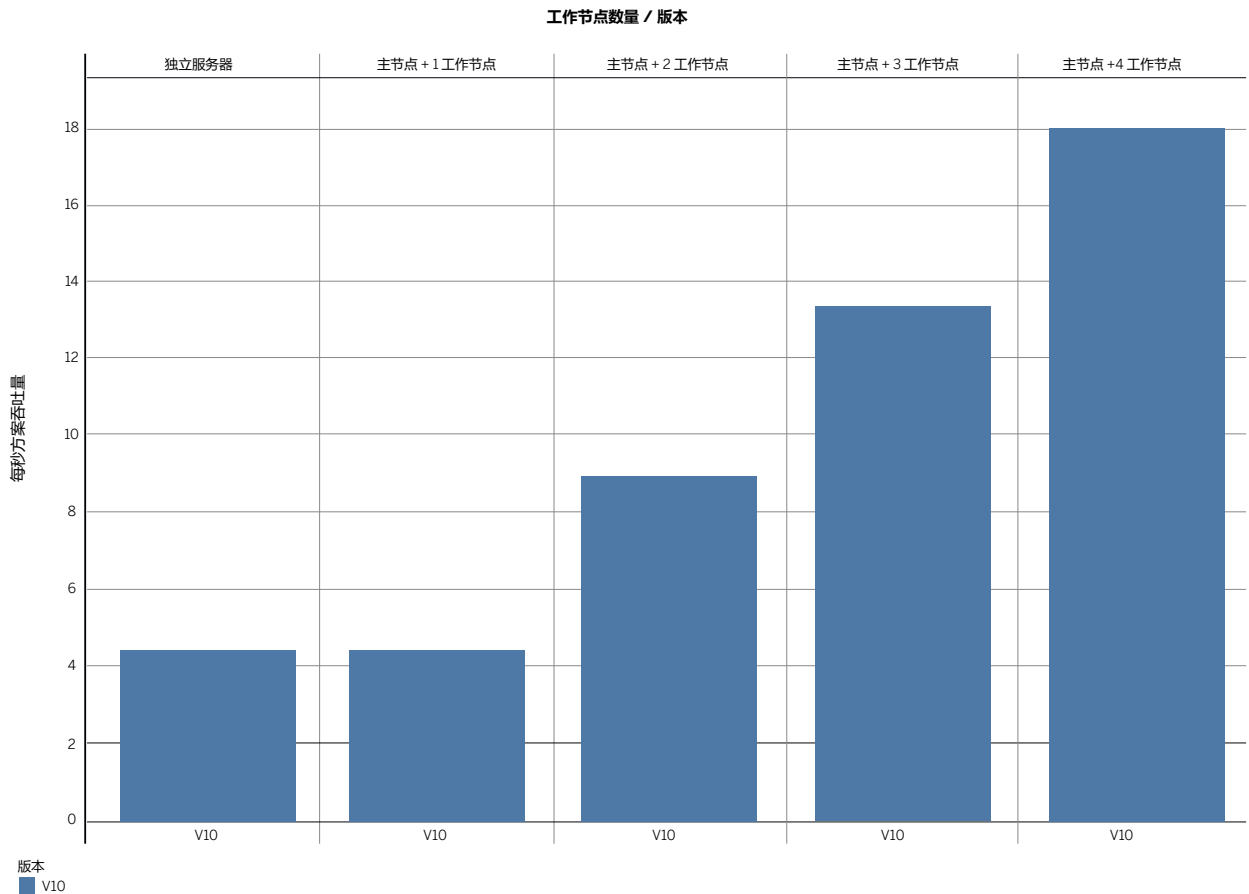


图 7：每秒方案吞吐量

每列显示了群集拓扑。最左侧的列显示了单服务器设置。第二列显示了具有单个工作节点的主/工作群集配置。我们在后面的每个列中添加一个工作节点，如前文所述。条形的高度显示每秒平均方案吞吐量 (TPS)。TPS 表示该服务器处理的工作量。如图所示，TPS 随工作节点的添加线性增加。我们观察到，当我们增加群集上的负载时，整个群集的 CPU 利用率平均值大约为 80% 下图以 95% 的置信区间显示了整个群集的 CPU 利用率在负载增加时的变化情况。

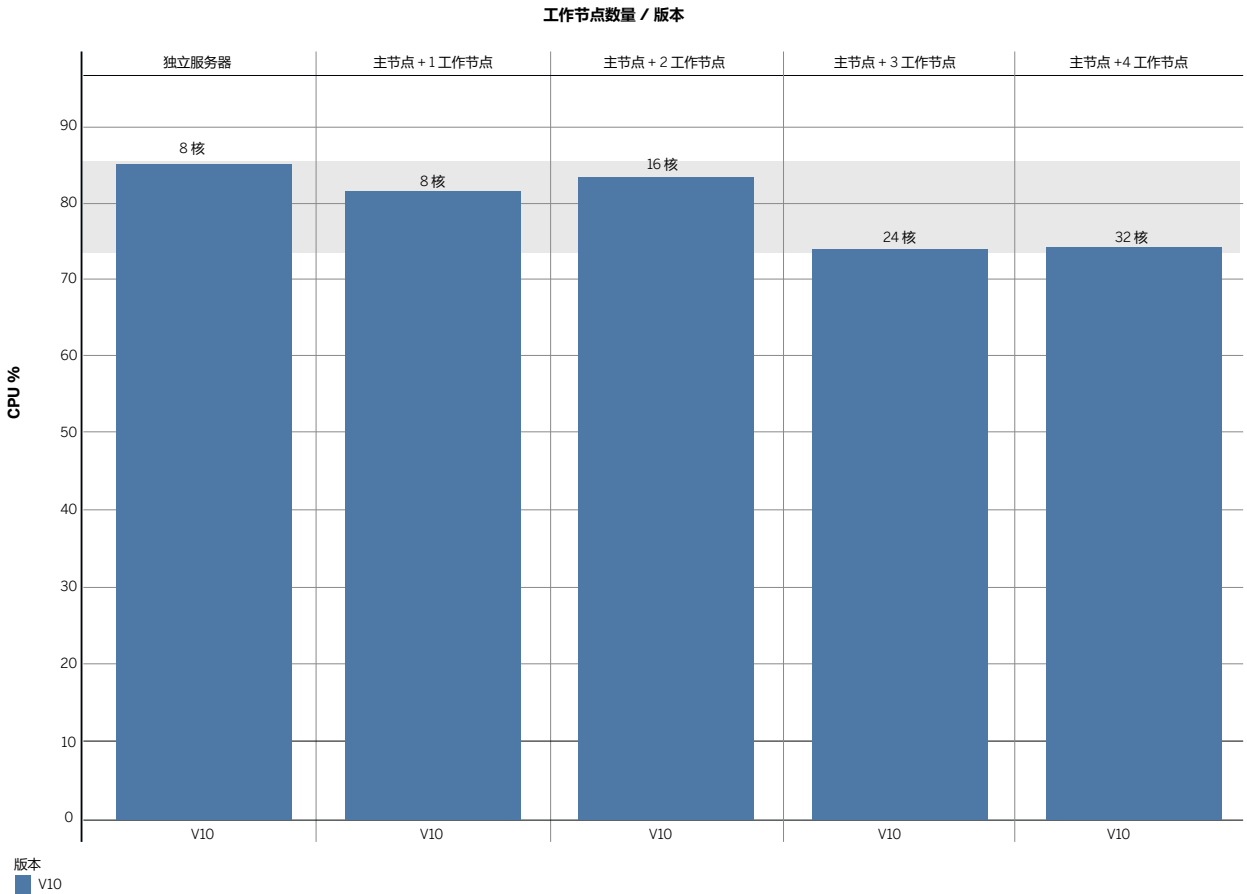


图 8：整个群集在负载增加时的 CPU 利用率

如图 8 所示，以添加更多工作节点的方式来分散整个群集的 CPU 负载可以为负载爆发提供储备空间，从而优化系统。与工作节点较多的群集相比，配置了较少工作节点数的群集有较高的 CPU 利用率。当工作节点数量较少时，需要计算资源的进程会相互竞争有限的资源。在这些测试的过程中，我们观察了 Server 的错误率，结果错误率被很好地控制在我们的方法体系所设置的 2% 的目标以内。如果群集包含较少的计算机且/或容量有限，您可能会看到更高的错误率（更低的服务质量），具体取决于您的工作负载。

我们的下一个问题是，在使用相同的方法体系时，Tableau Server 10 的表现与 Tableau Server 9.3 相比如何？

结果：对每个版本使用相同的方法体系时，与 Server 9.3 相比，Tableau Server 10 的吞吐量更高。

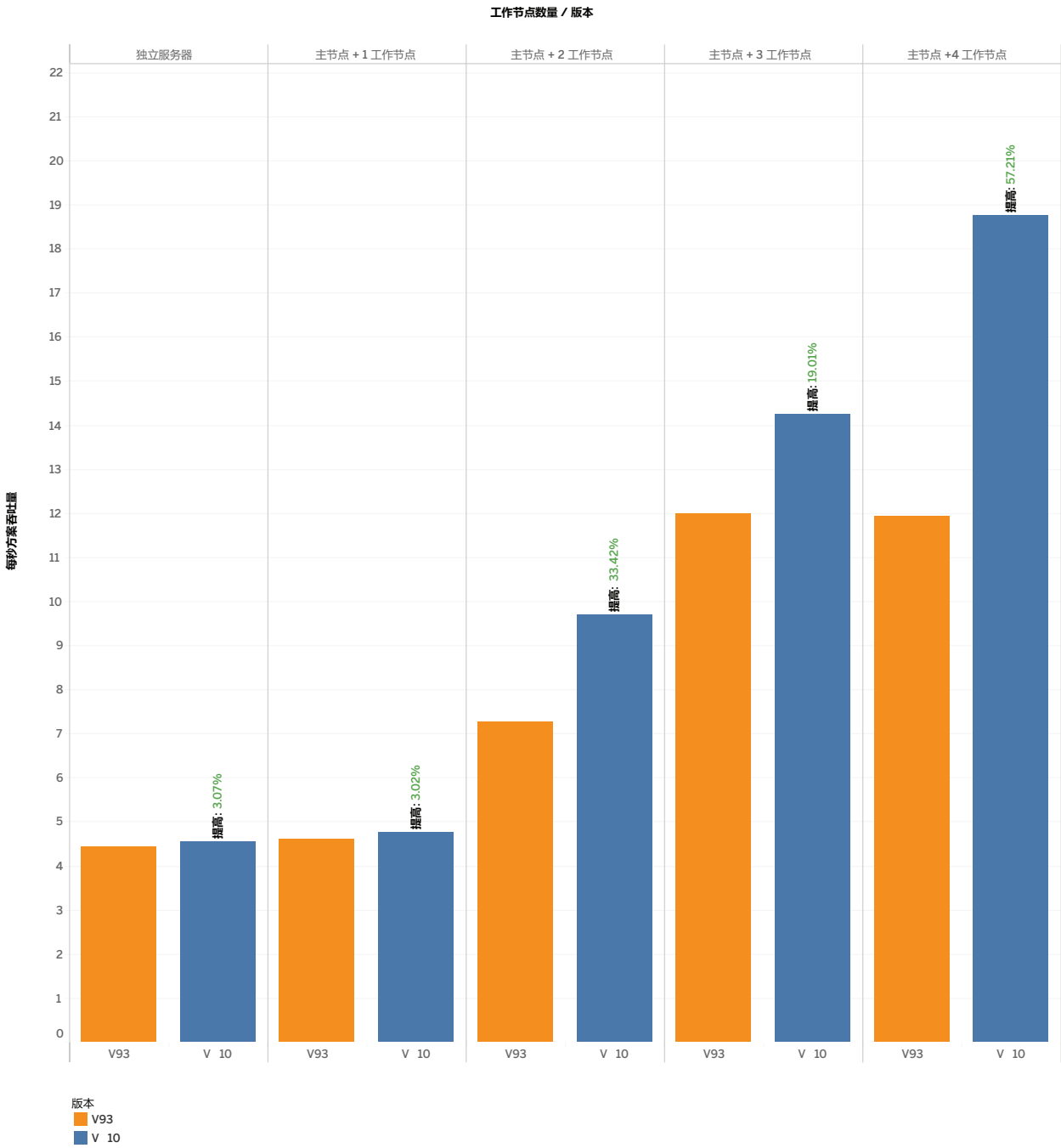


图 9 : 9.3 与 10.0.1 的每秒方案吞吐量对比

在图 9 中，我们对比了 Tableau Server 9.3 的方案吞吐量（橙色条形）与 Tableau Server 10 的方案吞吐量（蓝色条形）。每个窗格显示了列标题所描述的群集拓扑的配置。正如前文所提及，当我们将更多节点添加到群集时，Tableau Server 10 不仅实现了线性扩展，而且在方案吞吐量扩展方面的表现也优于 Tableau Server 9.3。

在对不同版本进行比较时，我们应该理解，唯一正确的比较方法是：采用经过改进的，能够体现真实用户方案的测试方法体系来对两种版本进行对比。正因为这样，将 Tableau Server 10 的结果与早期白皮书中的结果进行比较并不能准确反映二者的差异，因为测试方法体系已经发生了显著变化。

虽然观察吞吐量非常重要，但我们还希望确保整个测试方案的最终用户响应时间保持足够良好的状态，而不是在更高的负载下降低到标准以下。

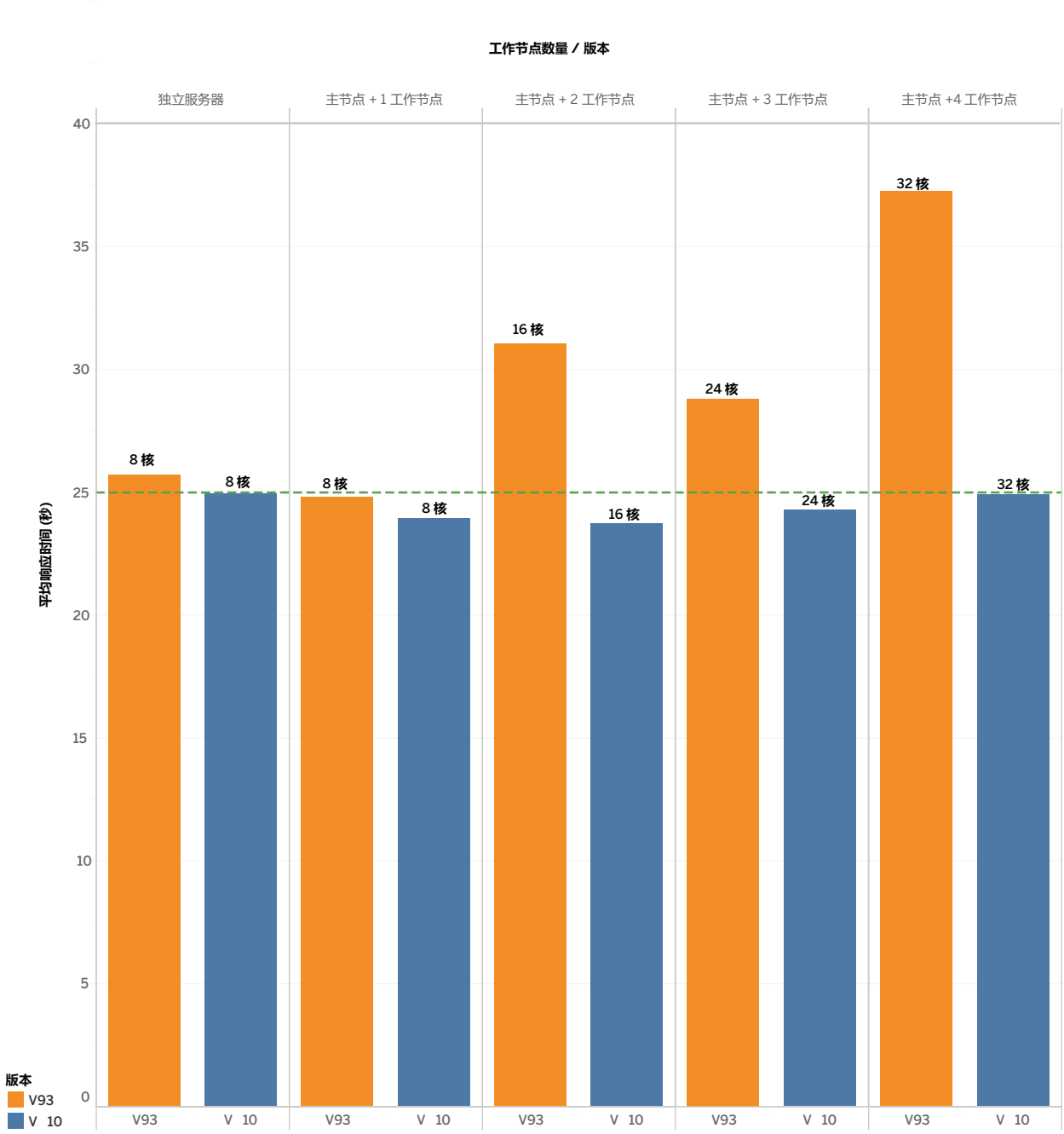


图 10：比较平均方案响应时间（秒）：9.3 与 10.0.1

在图 10 中，橙色条形的高度表示 9.3 的平均方案响应时间。蓝色条形的高度表示版本 10.0.1 的平均方案响应时间。当我们在托管版本 9.3 的群集上增加最终用户负载时，我们观察到响应时间不断递增。与此不同，当我们持续向系统添加虚拟用户时，Tableau Server 10 表现出更加稳定的响应时间。这表明，在工作负载相同时，Tableau Server 10 可以为最终用户提供优于 Tableau Server 9.3 的性能和响应能力。

Tableau Server 10 在负载增加时表现出相对一致和稳定的响应时间，这是因为我们改进了缓存组件。具体而言，我们确实在 v10 的开发工作中优化了引导响应方案的缓存。引导方案是为了启动用户会话并缓存会话数据而进行的第一次调用。先前的 Tableau Server 版本没有缓存功能，对于每个后续请求，它们都会计算和保存引导数据，即使这些数据非常相似甚至完全相同也不例外。在版本 10 中，我们对这一方案执行智能缓存操作。借助这些改进，Tableau Server 10 可以在保持良好响应时间的同时处理更多用户负载，从而变得更加高效。然而，正如我们此前在增加吞吐量时观察到的那样，Server 现在需要处理更多工作。

我们观察了 Tableau Server 在最终用户分析使用量增加时如何扩展，以上均为我们的观察结果。随着 Server 上最终用户工作负载的增加，您应该通过在群集中添加更多工作节点来确保足够的容量，以便为最终用户提供良好的服务质量体验。

以下一组问题的核心主题为后台程序的配置位置和方式将如何影响用户。具体而言，也就是如果将后台程序进程与分析服务 (VizQL Server) 托管在同一台计算机上，那么与将后台程序进程隔离到另一台计算机的配置相比，用户会受到怎样的影响。下一部分将对这些结果进行讨论。

后台程序结果

首先，我们想对在默认的单服务器安装中运行后台程序的影响进行定量。这种方案对最终用户以及总体可扩展性有何影响？我们在单个 Tableau Server 部署中运行了工作负载实验并记录了 TPS。随后，我们测试了群集上的工作负载，并通过添加工作节点来扩展群集拓扑。

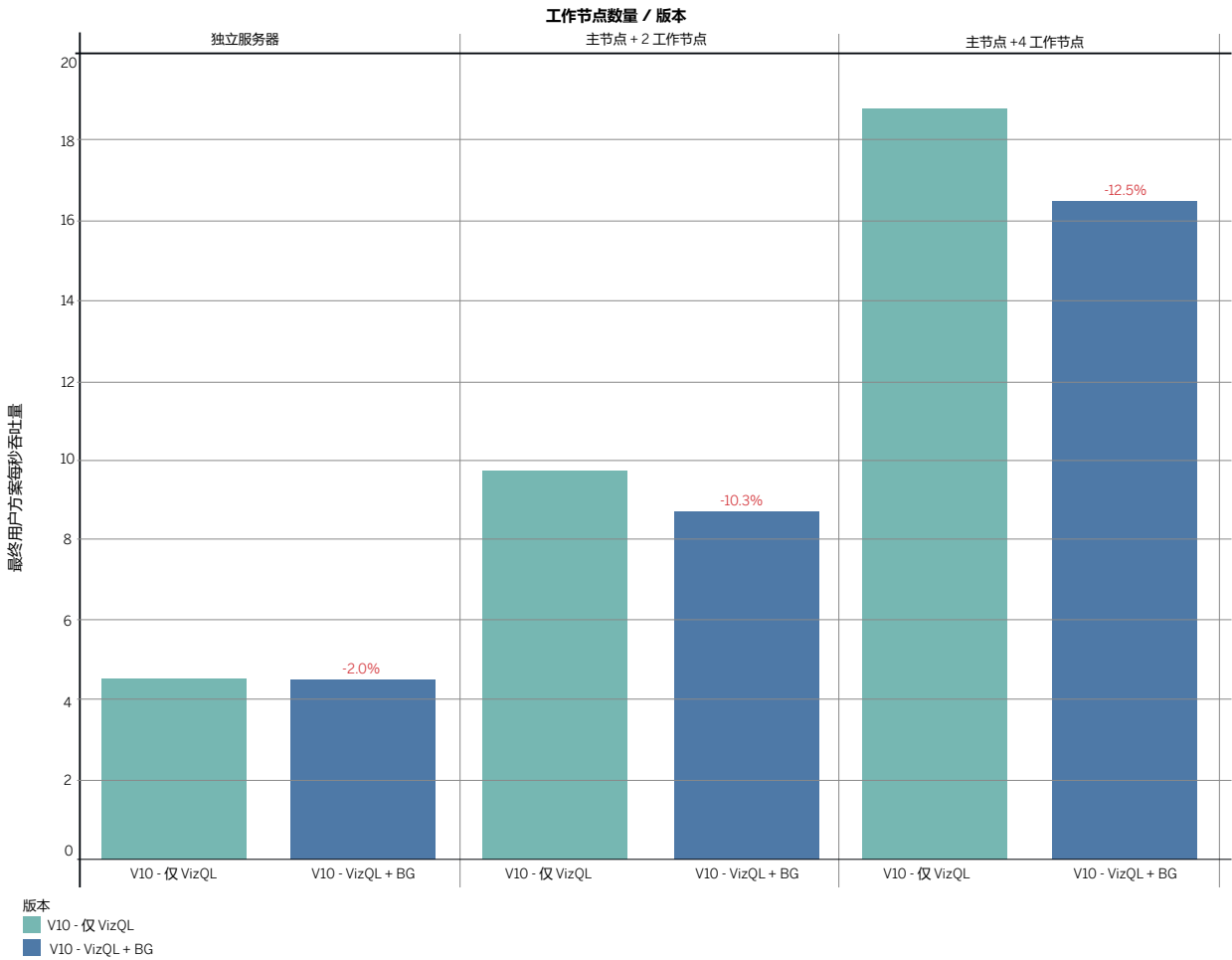


图 11：将后台程序和 VizQL 服务托管在同一位置对最终用户可扩展性的影响

图 11 显示了这两种测试方案的结果。绿色条形代表单独运行 VizQL 工作负载的测试方案。单独 VizQL 工作负载模拟最终用户工作负载。在蓝色条形代表的测试方案中，我们随后在 VizQL 工作负载的基础上，向同一个群集添加了固定数量的后台程序工作负载。随后，我们记录了两个方案中最终用户工作负载的 TPS 变化。我们已经预料到这会产生影响，因为 VizQL Server 和后台程序都是计算密集型工作负载，但我们的目的是测量具体工作负载的这种影响。

我们观察到最终用户方案吞吐量总体下降了 2-12%。这种下降表明，处理后台程序工作负载所需的资源消耗对群集为最终用户提供服务的能力造成了影响。根据这些数据可以推断，如果群集在单位时间内可以完成 100 个最终用户方案，那么在添加固定数量的后台程序负载后，相同单位时间的服务能力将降低为 88 个用户方案。吞吐量的这种降低可以体现为对最终用户可扩展性和/或服务质量的严重影响，具体取决于几个因素，例如工作负载、峰值负载爆发、硬件限制和架构变量。

这项测试说明，根据 Tableau Server 工作负载规划正确分配硬件资源是非常重要的。在功能不足或受限的硬件上运行 Tableau Server 可能导致吞吐量下降、后台程序作业失败、订阅通知延迟和/或最终用户错误，这些都会表现为性能问题。出现这种情况时，请考虑扩展群集，将后台程序工作负载隔离到专用硬件上。通过隔离后台程序服务，您可以释放争用计算资源的进程。如果这些进程位于同一台计算机上，它们将争夺同一批资源。

让我们分别针对后台程序与 VizQL Server 位于同一位置的配置，以及后台程序隔离到专用计算机上的配置，讨论具体的订阅工作负载受到的影响。

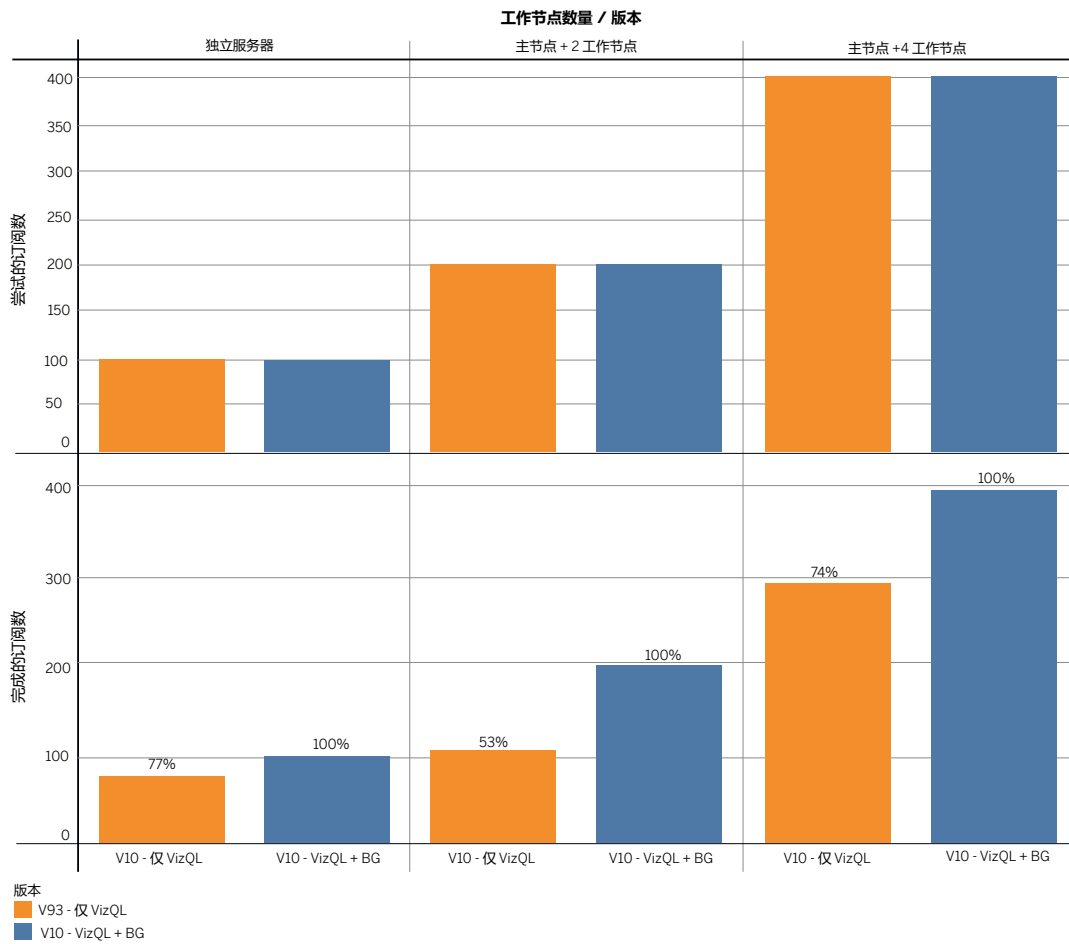


图 12：订阅通知改进

在图 12 中，每个窗格显示 9.3（橙色）和 10（蓝色）的 Tableau Server 群集拓扑，而每个条形代表订阅通知的尝试数量以及成功完成的通知数量。在每次迭代中，我们在步骤 0、2、4 增加工作节点数量，每个工作节点仅配置一个后台程序进程。（省略第 1 和第 3 个工作节点测试并不会影响测试结果的准确性。）对于每个工作节点，我们只配置一个后台程序进程，以便确立一致的测量标准，从而便于比较。

随着后台程序订阅负载的增加，Tableau Server 10 能够完成提交的所有作业。在同样的测试中，Tableau Server 9.3 会逐步饱和，一些订阅会失败。这种现象会在资源不足或负载过大的拓扑中进一步加剧。此外，我们观察到，将更多工作节点添加到系统后，Tableau Server 10 可以处理更多的订阅数量。这源于 Tableau Server 10 中一项改进，即为通知引入了映像缓存功能。借助这项功能，后台程序在执行与计划绑定的相同任务时，工作量大大降低。如果方案按照同样的计划执行同样的工作，Tableau Server 10 现在可以将第一次执行的结果放到缓存中，并为相同工作负载的后续请求提供这些结果。这意味着以更高的效率完成同样的工作。

对于按照相同的计划使用相同工作簿的订阅，我们看到，完成订阅所需的时间比 9.3 少 60-90%。图 13 显示了这种改进。

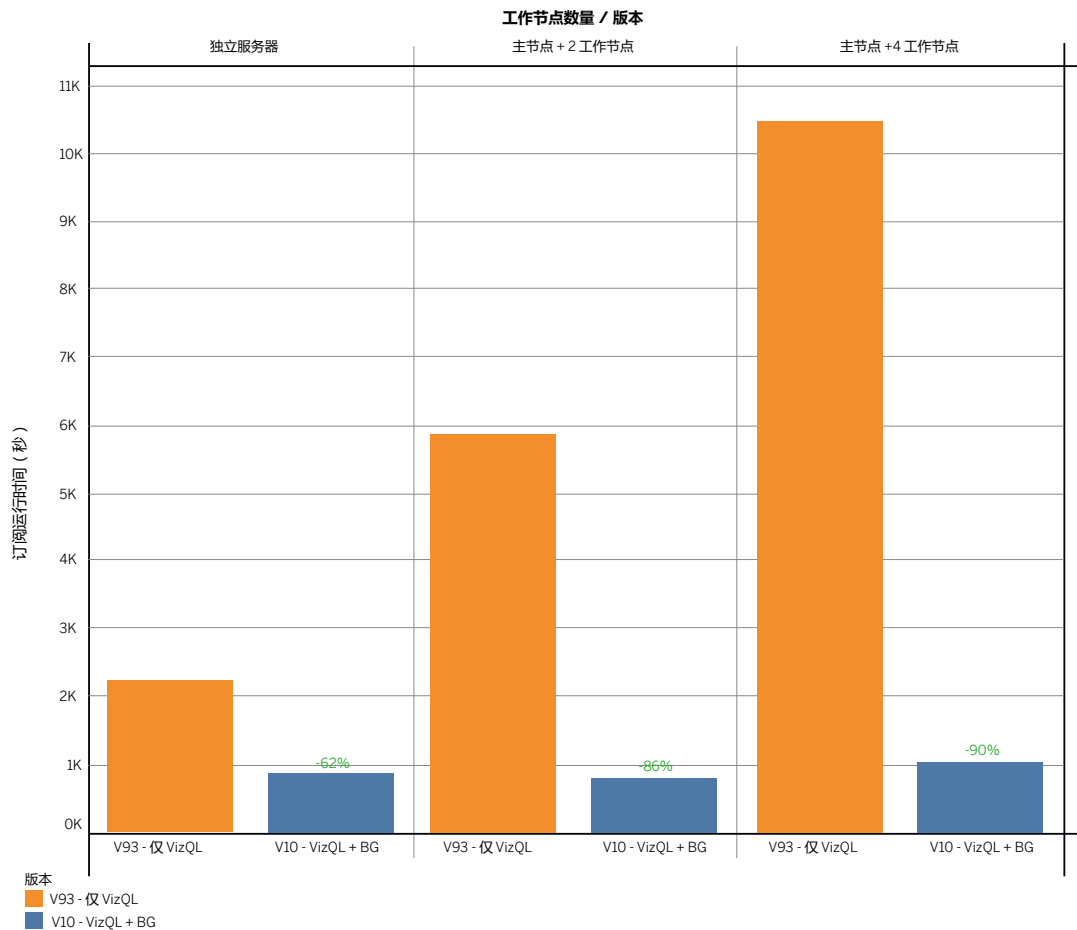


图 13：订阅运行时间（完成所需时间）：9.3 与 10

这意味着，按照同样的计划对同样的工作簿进行的任何订阅时，所需的完成时间将大大减少。但如有不同工作簿的通知，或者设置了用户筛选器，则在向用户发送通知前，订阅需要一些时间来进行处理和计算。不同的工作簿和用户筛选器要求 Tableau Server 运行整个最终用户可视化管道，包括每个工作簿或筛选视图的数据查询和数据可视化进程。

订阅的好处在于，它们能够及时为业务用户提供他们关心的数据。有效的数据提取刷新可以帮助您的组织根据具有适宜新鲜度的数据制定明智的决策。因为后台程序进程同时管理这两个关键功能，因此，通过对订阅计划进行规划，并将重复的工作绑定到同样的计划，就可以确保缓存的优势得到发挥。

隔离后台程序进程

在将后台程序进程隔离到其专用的工作节点后，我们运行了同样的订阅实验。我们观察到，单个 4 核计算机上的后台程序工作节点能够完成 400 个订阅。当我们在 4 台工作计算机上分别托管单个后台程序进程与单个 VizQL 时，所记录的订阅总数与此相等。

我们从这里学到的重要知识是：虽然后台程序本身可以扩展，隔离该进程可以在不影响最终用户服务质量的情况下实现相似的扩展。后台程序是单线程进程，用于以最快的速度完成作业。基于这种设计，后台程序进程在处理工作任务时将占用整个核心。在一台隔离的计算机上，后台程序进程对计算资源的掠夺式使用将不会影响其他面向用户的 Tableau 服务。

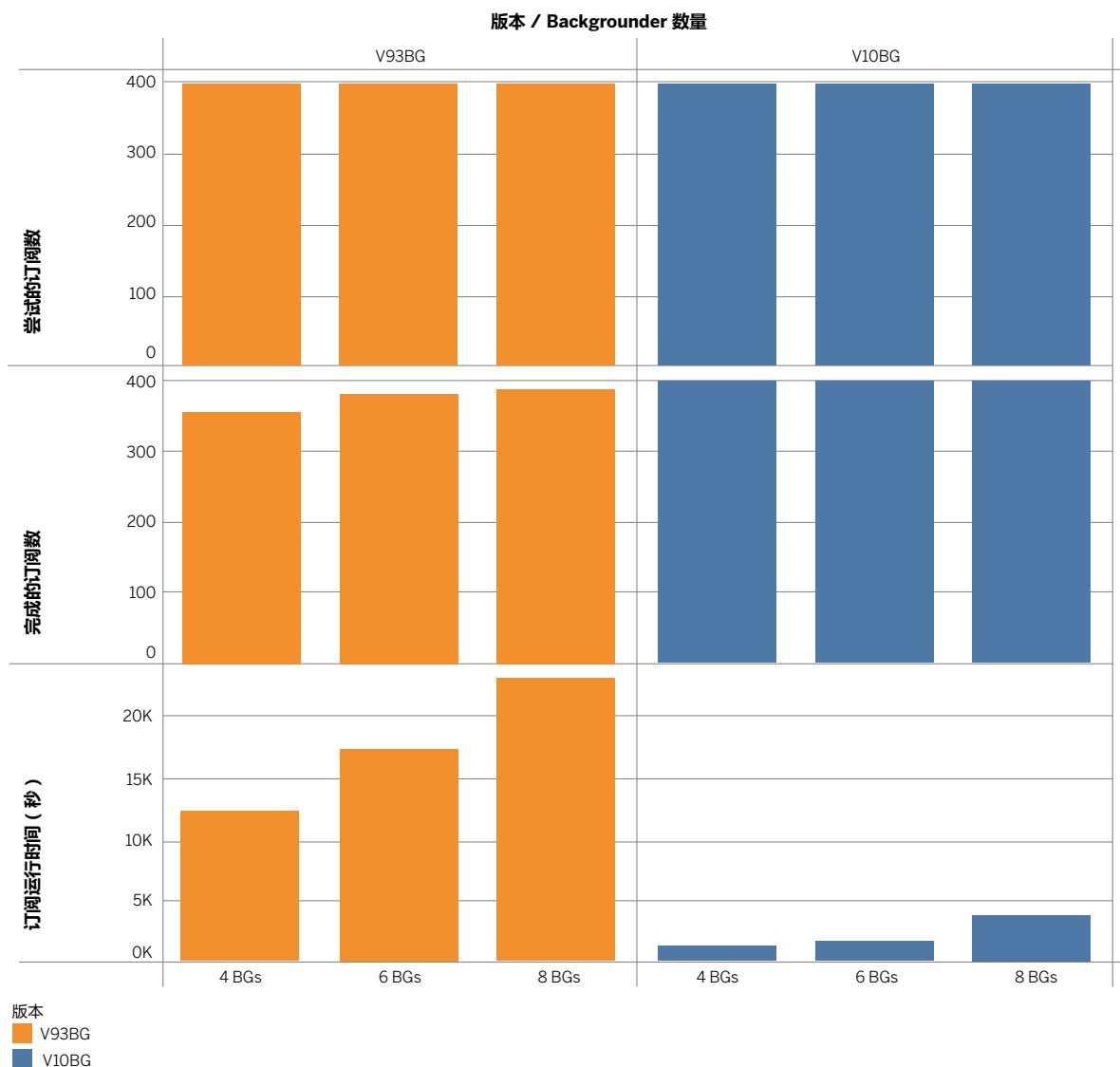


图 14：在 4 核计算机上添加后台程序进程：9.3 与 10

如图 14 所示，在单个 4 核计算机上添加后台程序进程让 Tableau Server 10（蓝色）能够在显著低于 9.3（橙色）的时间内完成订阅。然而，如果计算机因为核心配置而受到物理限制，则继续添加后台程序进程将会产生负面影响。如图所示，在受 4 核限制的计算机上添加 8 个后台程序进程将增加完成时间。这是后台程序的单线程设计造成的。

最后，我们对运行隔离后台程序进程的工作节点进行了数据提取刷新测试，在这些实验中，我们观察到 Server 9.3 与 Server 10 的表现大致相同。两个版本在大致相同的时间内完成了相同数量的数据提取刷新。针对数据提取刷新进行扩展时，应该考虑的一个重要细节是：数据提取刷新良好性能高度依赖于外部数据库。（在我们的测试中，我们用来进行数据刷新的工作簿使用了来自 MS SQL Server 的已发布数据提取。）数据提取刷新的性能和可扩展性高度依赖于数据库硬件规格。此外，数据特征（例如联接类型以及所执行的查询的复杂度）也会影响扩展。因此，您应该确保数据提取刷新和通知具有足够多的可用容量，以便在系统出现最终用户峰值负载的预期时间前完成其工作。

后台程序注意事项

后台程序进程会执行很多与数据提取刷新、订阅以及其他计划的后台作业相关的工作。如果您安排这些作业在非高峰时段运行，它们就不会争用容量。如果无法这样安排，则应该对后台程序和其他并非面向用户的工作负载进行规划，并为其添加与面向用户的进程同时运行所需的容量。

根据设计，每个后台程序进程会占用整个核心的容量，因为它们的宗旨是尽快完成工作。运行多个后台程序进程时，应该考虑以下事实：后台程序进程可能与在同一台计算机上运行的其他服务争夺计算和网络资源。

最佳做法 – 自行进行扩展测试

如果您希望执行自己的负载测试，以确定 Tableau Server 在自己的环境中如何随工作负载进行扩展，不妨参考下文提供的一些最佳做法。

- 1. 不要把 Tableau Server 当成黑盒。**传统负载测试往往将被测试的应用程序视为黑盒。这会假设不能根据负载条件来对部署进行调优/配置或调整。Tableau 采用了易于进行纵向和横向扩展的设计，理解 Tableau 架构有助于为可扩展性测试提供有用信息，从而获得适用于您具体情况的结果。
- 2. 选择正确的测试工具。**Tableau Server 拥有强劲动力，可承担各种复杂的资源密集型工作。有很多工具可以用来在 Tableau Server 上生成负载。虽然 Tableau 不直接支持其中的任何工具，但还是应挑选出最易用且最贴近您生产环境的工具。另一个考虑事项是，确保您具备相应工具和 Tableau Server 方面的相关专业知识以进行负载测试。我们使用 **Tabjolt** 进行测试。Tabjolt 是一款基于 Jmeter 的“即点即运行”负载测试工具，它可以在用于 Tableau 等临时分析解决方案时免除脚本维护工作。
- 3. 甄选有代表性的工作簿。**很多时候，我们之所以会听到性能或扩展方面的问题，是因为我们使用的工作簿并非遵循最佳做法制作。如果针对工作簿的单一用户测试结果表明响应时间极长，则应该优化这些工作簿，然后才能着手开展负载测试项目。您不希望在生产环境中保留性能低下的仪表盘。同样，您也不会希望将其用于测试。
- 4. 首先使用默认配置。**通过实时连接测试工作簿时，请注意一点，Tableau Server 9.0 中已引入并行化功能，因此需要的 VizQL 服务器数量会少于您在先前 Tableau Server 版本中可能部署的数量。首先使用新的双进程默认配置，然后根据需要逐步扩展。

实际环境优化最佳做法

除了经过优化设计的系统之外，还有一些最佳做法可以显著提升性能，缩短平均响应时间。

- 设计外观和性能俱佳的工作簿。很多时候，我们之所以会听到客户说自己的工作簿太慢，是因为性能没有在工作簿的设计过程中得到重视。如果单用户负载时间很长，该工作簿的响应时间在高负载下也会很长。在构建分析文化的同时，您可以通过提供相应的途径和组建相应的团队，让用户在“管家”的帮助下设计出外观漂亮、见解丰富、性能优良的工作簿。这种做法同样可以确保您构建和提供可扩展的可视化。“[设计高效工作簿](#)”是一份白皮书，该白皮书更加深入地介绍了如何构建性能良好的高效仪表板。
- 最终用户体验的合计响应时间包含了多个组成部分，但最主要的是 Tableau 消耗的时间和数据检索时间。如果后端数据库速度慢，或者查询时间长，可视化也会比较慢。考虑自己的数据策略非常重要。组织中的数据源往往是经过整理的共享数据源。您必须确保自己提供可以提高业务用户工作效率的重要数据。这意味着优化数据。例如，您应该确保最佳的联接以及相关的聚合级别，以便对索引表进行快速检索。要使可视化保持良好的性能，良好的数据清理流程非常重要。
- 使用 Tableau 数据提取。如果数据库查询较慢，可考虑使用数据提取来提高查询性能。数据提取存储在服务器本地并在内存中运行，因此用户无需向数据库提出请求就可以快速访问这些数据。数据提取可以轻松进行筛选和聚合，对于不需要行级别详细信息的用户而言，堪称理想的选择。数据提取可以显著改善响应时间，让用户进入连贯的分析流。
- 将更新安排在非峰值时段。数据源通常会实时更新，但用户只是每天或每周才需要一次数据。将数据提取安排在非高峰时段，可以减轻数据库和 Tableau Server 在高峰时段的负载。不仅如此，还可以在现有计算机上添加更多后台程序，或者使用专用硬件，但前提是您有充足的核心容量。如果希望以更快的速度完成数据提取，不妨考虑此方案。
- 在峰值时段避免“奢侈”操作。文件发布（特别是发布大文件时）是非常消耗资源的任务。改变发布行为往往很简单：让用户在非峰值时段进行发布，避开繁忙时段，例如星期一早上。要了解服务器何时的使用量最大，请使用[管理员视图](#)，然后根据实际使用量制定策略。根据您对 Tableau Server 10.0 所采用的配置方式，发布内容也意味着会在各群集节点上创建一份数据提取副本，以实现高可用性。通过非高峰时段执行此操作，还能让您最大程度利用网络带宽。
- 缓存视图。当有多个用户开始访问 Tableau Server 时，由于会争用共享资源，响应时间最初会增加。开启缓存功能之后，进入系统的所有请求的视图都会被缓存，然后以更快的速度呈现给同一仪表板的下一位查看者。
- 缓存服务器进程是 Tableau Server 9.0 引入的一项功能，可通过如下方式进行预热：计划在完成数据提取刷新后发送普通视图电子邮件。这样一来，后续查看者使用的就是系统为之前的请求缓存的数据。您可以使用其他方法预热缓存，例如借助自动化工具加载访问量通常较大的重要可视化。用户可以随时手动取消外部查询缓存，以便从数据源刷新数据。此操作还会强制重新生成缓存。这样一来，无论缓存中是否已有版本，用户总能获得一份最新的数据。

总结

Tableau Server 10 是一个可扩展的企业级平台，可以支持任何规模的组织。它可以在本地、私有云或公有云中运行，并可以随工作计算机容量的增加实现线性扩展。每个环境都会有自己独有的特征和配置，但 Tableau Server 的架构让您总是能够通过扩展部署来满足用户需求。

您的可扩展性和性能指标可能与此不同，并且我们的建议并不是适用于所有情况的具体建议，但 Tableau Server 10 可以使用 8-16 核的容量支持包含 25 至 100 名用户的团队或部门。如果您打算支持包含 100 至 1000 名用户的团队，根据使用量和数据新鲜度需求，16 至 24 核可能是不错的起点配置。如果需要支持更多用户或更多后台负载，您可以将自己的部署增加到 32-64 核以上，通过添加更多工作节点来进行扩展（甚至扩展到云规模），以支持不断增加的工作负载。

关于 Tableau

Tableau 帮助人们将数据转化为可以付诸行动，发挥重大作用的见解。轻松连接到以任何形式存储在任意地点的数据。快速执行临时分析，发现隐藏的见解。通过拖放操作，创建包含高级可视化分析的交互式仪表板。然后在整个组织共享，让其他团队成员能够从自己的数据视角进行探索。从全球性企业到早期初创企业和小企业，使用 Tableau 的分析平台来查看和理解数据的人无处不在。

资源

[Tableau for the Enterprise: An IT overview \(适用于企业的 Tableau : IT 概述\)](#)

[Server 管理指南](#)

[Tableau Server 10.0 高可用性：大规模提供任务关键型分析](#)

[Tableau on Amazon Web Services \(基于 Amazon Web Services 的 Tableau\)](#)

