



Informatica®
10.4.1

性能优化指南

Informatica 性能优化指南

10.4.1

2020 年 6 月

© 版权所有 Informatica LLC 2009, 2020

本软件和文档仅根据包含使用与披露限制的单独许可协议提供。未事先征得 Informatica LLC 同意，不得以任何形式、通过任何手段（电子、影印、录制或其他手段）复制或传播本文档的任何部分。

Informatica 和 Informatica 标志是 Informatica LLC 在美国和世界其他许多司法管辖区的商标或注册商标。欲获得 Informatica 商标的最新列表，请访问 <https://www.informatica.com/trademarks.html>。其他公司和产品名称可能是其各自所有者的商业名称或商标。

美国政府权利交付给美国政府客户的程序、软件、数据库及相关文档和技术数据是指适用的联邦采购条例和政府机构特定补充条例中定义的"商业计算机软件"或"商业技术数据"。因此，使用、复制、披露、修改和改编应遵循适用的政府合同中规定的限制和许可条款、政府合同条款的适用范围以及 FAR 52.227-19 商用计算机软件许可中规定的额外权利。

本软件和/或文档中的若干部分受第三方版权约束。所需的第三方声明随产品一起提供。

本文档中的信息如有更改，恕不另行通知。如发现本文档中有什么问题，请通过以下电子邮件地址向我们报告：infa_documentation@informatica.com。

Informatica 产品根据对应协议的条款和条件进行担保。INFORMATICA 按"原样"提供本文档中的信息，无任何明示或暗示的担保，包括但不限于任何适销性和特定用途适用性担保，也没有任何非侵权担保或条件。

发布日期: 2020-08-04

目录

前言	7
Informatica 资源	7
Informatica Network	7
Informatica 知识库	7
Informatica 文档	7
Informatica 产品可用性矩阵	8
Informatica Velocity	8
Informatica Marketplace	8
Informatica 全球客户支持部门	8
第 1 章：性能优化概览	9
性能优化概览	9
性能优化过程	9
目标瓶颈	10
源瓶颈	10
映射瓶颈	10
计算机系统瓶颈	11
标识 Windows 中的系统瓶颈	11
标识 UNIX 中的系统瓶颈	11
运行时瓶颈	11
连接瓶颈	12
第 2 章：目标优化	13
目标优化概览	13
平面文件目标优化	13
数据库检查点时间间隔	14
批量加载	14
数据库目标优化	14
第 3 章：源优化	15
源优化概览	15
平面文件源优化	15
查询优化	16
条件筛选器	16
选择相异	17
提示	17
提示规则和准则	18
创建提示	18
约束	18
配置约束	19

自定义数据对象优化.	19
数据库源优化.	20
第 4 章：转换优化.	21
转换优化.	21
汇总器转换优化.	21
表达式优化.	22
Java 转换优化.	24
通过 Java 转换执行早期选择优化.	24
通过 Java 转换执行推入优化.	25
联接器转换优化.	26
查找转换优化.	26
排序器转换优化.	28
SQL 转换优化.	29
通过 SQL 转换执行早期选择优化.	29
通过 SQL 转换执行推入优化.	29
转换缓存.	30
转换错误消除.	31
转换的副作用.	31
Web 服务使用者转换优化.	31
通过 Web 服务使用者转换执行早期选择优化.	32
通过 Web 服务使用者转换执行推入优化.	32
第 5 章：映射优化.	34
映射优化概览.	34
优化方法.	35
优化器级别.	35
筛选器优化.	36
早期计划优化方法.	36
谓词优化方法.	36
基于成本的优化方法.	37
船舶数据中心联接优化方法.	37
半联接优化方法.	38
早期选择优化方法.	39
全局谓词优化方法.	39
分支修剪优化方法.	39
推入优化方法.	40
下推优化.	40
完整下推优化.	40
源下推.	41
下推优化规则和准则.	41
一次性读取.	42
筛选器优化.	42

数据类型转换优化.	42
错误跟踪.	43
第 6 章：已分区映射优化.	44
已分区映射优化概述.	44
使用多个 CPU.	44
增加最大并行数值.	45
针对分区优化平面文件.	45
针对分区优化平面文件源.	45
针对分区优化平面文件目标.	45
优化分区的关系数据库.	46
优化源数据库以进行分区.	46
优化目标数据库以进行分区.	47
针对分区优化转换.	47
第 7 章：运行时优化.	49
运行时优化概览.	49
监视统计信息.	49
内存分配.	50
数据对象缓存.	51
缓存表的数据类型.	51
数据对象缓存优化.	53
系统优化.	53
第 8 章：SQL 数据服务优化.	55
SQL 数据服务优化概览.	55
第三方客户端工具优化.	55
SQL 数据服务优化器级别.	56
为数据预览配置 SQL 数据服务优化器级别.	57
为已部署的 SQL 数据服务配置优化器级别.	57
SQL 数据服务查询计划.	57
查看 SQL 查询计划.	58
SQL 数据服务的内存和并发请求属性.	59
SQL 数据服务的结果集缓存.	60
SQL 数据服务结果集缓存属性.	60
启用 SQL 数据服务的结果集缓存.	61
在临时表中保留虚拟数据.	61
实施临时表.	61
第 9 章：Web 服务优化.	62
Web 服务优化概览.	62
优化 HTTP 请求.	63
Web 服务消息压缩.	63

Web 服务优化器级别.	63
为数据预览配置 Web 服务优化器级别.	64
为已部署的 Web 服务配置优化器级别.	64
内存和并发请求的 Web 服务属性.	65
并发 Web 服务请求的数据集成服务配置示例.	66
用于配置活动 DTM 实例的 Web 服务属性.	67
Web 服务结果集缓存.	67
为 Web 服务启用结果集缓存.	67
Web 服务日志管理.	68
 第 10 章：连接优化.	 69
连接优化概览.	69
连接池.	69
连接对象中的池属性.	69
数据库网络包大小.	70
 索引.	 71

前言

要学习如何优化映射性能，请阅读《*Informatica® 性能优化指南*》。了解如何发现和消除每个映射组件中的性能瓶颈。

Informatica 资源

Informatica 通过 Informatica Network 和其他在线门户为您提供一系列产品资源。使用这些资源，可以充分利用 Informatica 产品和解决方案，并向其他 Informatica 用户和主题专家学习。

Informatica Network

在 Informatica Network 中可以获得许多资源，包括 Informatica 知识库和 Informatica 全球客户支持。要进入 Informatica Network，请访问 <https://network.informatica.com>。

作为 Informatica Network 成员，您可以选择以下服务：

- 在知识库中搜索产品资源。
- 查看产品可用性信息。
- 创建并检查您的支持案例。
- 查找当地的 Informatica 用户组网络并与您的伙伴进行协作。

Informatica 知识库

使用 Informatica 知识库可查找产品资源，例如操作方法文章、最佳实践、视频教程以及常见问题的答案。

要搜索知识库，请访问 <https://search.informatica.com>。如果您对知识库有任何疑问、意见或建议，请与 Informatica 知识库团队联系，电子邮件地址为 KB_Feedback@informatica.com。

Informatica 文档

使用 Informatica 文档门户可浏览大量当前与最近产品版本的文档库。要浏览文档门户，请访问 <https://docs.informatica.com>。

如果您对产品文档有任何疑问、意见或建议，请与 Informatica 文档团队联系，电子邮件地址为 infa_documentation@informatica.com。

Informatica 产品可用性矩阵

产品可用性矩阵 (PAM) 指明了产品版本支持的操作系统版本、数据库以及数据源和目标的类型。您可以在以下网址中浏览 Informatica PAM:

<https://network.informatica.com/community/informatica-network/product-availability-matrices>。

Informatica Velocity

Informatica Velocity 是由 Informatica 专业服务根据数百个数据管理项目的实际经验所开发出来的，其中汇集了大量使用技巧和最佳实践。Informatica Velocity 代表了 Informatica 顾问的集体知识，这些顾问与世界各地的组织合作，共同计划、开发、部署和维护成功的数据管理解决方案。

您可以在以下网址中找到 Informatica Velocity 资源：<http://velocity.informatica.com>。如果您对 Informatica Velocity 有任何疑问、意见或建议，请通过 ips@informatica.com 与 Informatica 专业服务联系。

Informatica Marketplace

Informatica Marketplace 是一个论坛，该论坛中提供的解决方案可扩展和增强您的 Informatica 实施。利用 Informatica 开发人员和合作伙伴在 Marketplace 中提供的数以百计的解决方案，可提高您的工作效率并加快项目实施时间。您可以在以下网址中找到 Informatica Marketplace：<https://marketplace.informatica.com>。

Informatica 全球客户支持部门

您可以通过电话或 Informatica Network 与全球支持中心联系。

要查找您当地的 Informatica 全球客户支持部门电话号码，请访问 Informatica 网站，链接为：<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>。

要在 Informatica Network 上查找在线支持资源，请访问 <https://network.informatica.com>，然后选择 eSupport 选项。

第 1 章

性能优化概览

本章包括以下主题：

- [性能优化概览, 9](#)
- [性能优化过程, 9](#)
- [目标瓶颈, 10](#)
- [源瓶颈, 10](#)
- [映射瓶颈, 10](#)
- [计算机系统瓶颈, 11](#)
- [运行时瓶颈, 11](#)
- [连接瓶颈, 12](#)

性能优化概览

消除性能瓶颈是性能优化的目标。瓶颈是指运行最频繁但吞吐量最低的映射中的区域。瓶颈可降低映射的整体性能。

要优化映射，请标识一个性能瓶颈、消除该瓶颈，然后标识下一个性能瓶颈。一次请优化一个映射组件。您可以在更改前和更改后对映射进行计时，以验证优化是否对性能有影响。

可以查找并修复诸如源、目标和连接等映射组件中的瓶颈。可以检查数据集成服务和运行数据集成服务的计算机中的瓶颈。还可以优化 Web 服务和 SQL 数据服务的属性。

性能优化过程

可以按照一组步骤来优化映射组件并提高性能。

您可以按以下顺序优化映射组件：

1. 目标
2. 源
3. 映射
4. 转换
5. Administrator 工具中的 Informatica 环境

6. 计算机系统

7. 数据服务或 Web 服务

使用以下方法可标识性能瓶颈：

- 运行测试映射。您可以配置一个从平面文件源中读取数据或将数据写入平面文件目标的测试映射，以标识源和目标瓶颈。
- 分析性能详细信息。分析性能详细信息（例如优化方法）以确定映射性能降低的位置。
- 监视系统性能。您可以使用系统监视工具来查看 CPU 使用百分比、I/O 等待时间、分页以及系统资源使用率。

目标瓶颈

数据集成服务向目标写入数据时，目标瓶颈将降低性能。数据库使用小检查点时间间隔或小数据库网络包大小时，可能会出现目标瓶颈。

数据集成服务向目标数据库写入数据时，将出现最常见的性能瓶颈。如果数据库使用小检查点时间间隔，则在写入小检查点时数据库处理速度会更加频繁地降低。小数据库网络包大小可能会导致出现瓶颈。您可以允许更大的数据包同时跨网络。

要标识目标瓶颈，您可以创建包含平面文件目标而非数据库目标的映射的备份。如果性能显著提高，您即遇到目标瓶颈。如果映射已写入平面文件目标，您可能没有遇到目标瓶颈。

源瓶颈

数据集成服务从源数据库中读取数据时，源瓶颈会降低性能。如果源查询效率不高，或者如果数据库网络包大小非常小，则可能会出现源瓶颈。

映射从关系源中读取数据时，您可以使用以下方法标识源瓶颈：

- 将筛选器转换添加到映射中。在源后添加筛选器转换。将筛选条件设置为 false，以使筛选器转换不返回任何数据。如果映射所需的时间大致相同，则映射具有源瓶颈。
- 创建读取测试映射。备份映射但不删除所有转换、联接或查询。将源连接到目标。如果映射性能与原始映射相似，您即遇到源瓶颈。
- 直接针对源数据库运行读取查询。从映射日志中复制读取查询。使用查询工具（例如 isql）针对源数据库运行查询。测量运行时间以及查询返回一行所需的时间。

映射瓶颈

如果您确定自己没有遇到源或目标瓶颈，则可能会遇到映射瓶颈。缓存大小较小、缓冲区内内存较低以及提交间隔较短可能会导致出现映射瓶颈。

要标识映射瓶颈，请分析映射日志中的性能详细信息。性能详细信息包括与每个转换有关的信息，例如输入行数、输出行数和错误行数。

还可以在目标定义前添加筛选器转换。将筛选条件设置为 false，以使筛选器转换不会将任何数据加载到目标表。如果运行新映射所需的时间与原始映射相同，您即遇到映射瓶颈。

计算机系统瓶颈

在 Windows 或 UNIX 中运行 Informatica 服务时，您可以查看资源使用率。在 Windows 中，请使用任务管理器。UNIX 有多种可用于查看性能的工具。

标识 Windows 中的系统瓶颈

您可以查看任务管理器中的“性能”和“进程”选项卡，了解系统信息。任务管理器中的“性能”选项卡提供 CPU 使用率和使用的总内存概览。使用性能监视器可查看更多信息。

下表介绍了 Windows 性能监视器中可用于创建图表的系统信息：

属性	说明
处理器时间百分比	如果您有多个 CPU，请监视每个 CPU，了解处理器时间百分比。
Pages/second	如果 pages/second 大于 5，您的内存压力可能过大（称为抖动）。
物理磁盘时间百分比	物理磁盘忙于执行读取或写入请求的时间百分比。
物理磁盘队列长度	等待访问同一磁盘设备的用户数。
每秒服务器总字节数	服务器已发送到网络以及从中接收的数据量。

标识 UNIX 中的系统瓶颈

使用以下工具可标识 UNIX 中的系统瓶颈。

- top. 查看整体系统性能。此工具显示系统以及系统中运行的单个进程的 CPU 使用率、内存使用率以及交换区使用率。
- iostat. 监视连接到数据库服务器的每个磁盘的加载操作。iostat 显示磁盘实际活动的时间百分比。如果您使用磁盘阵列，请使用随磁盘阵列提供的实用工具，而非 iostat。
- vmstat. 监视磁盘交换操作。
- sar. 查看 CPU、内存和磁盘使用率的详细系统活动报告。您可以使用此工具监视 CPU 加载。此工具提供与用户、系统、空闲时间和等待时间有关的百分比使用率。还可以使用此工具监视磁盘交换操作。

运行时瓶颈

要优化映射性能，请启用性能功能并调整数据集成服务属性。请在 Administrator 工具中为数据集成服务和模型存储库服务配置优化设置。

请分配实现最佳系统性能所需的内存并配置错误跟踪级别，以减少数据集成服务在运行映射时生成的日志事件数量。

您可以配置数据集成服务为运行所有并发请求而分配的内存量上限。还可以限制数据集成服务为任何给定请求分配的内存量上限。

您可以配置结果集缓存，以使数据集成服务能够缓存与每个 SQL 数据服务查询和 Web 服务请求关联的 DTM 进程的结果。

连接瓶颈

您可以优化连接以提高性能。您可以管理数据库连接的空闲连接实例的池。您可以增大网络包大小以允许更大型的数据包同时跨网络。

第 2 章

目标优化

本章包括以下主题：

- [目标优化概览, 13](#)
- [平面文件目标优化, 13](#)
- [数据库检查点时间间隔, 14](#)
- [批量加载, 14](#)
- [数据库目标优化, 14](#)

目标优化概览

优化目标可使数据集成服务能够有效地写入目标。您可以在运行映射之前删除索引和关键约束、增加数据库中检查点时间间隔的数量、在数据对象的写入属性中配置批量加载以及优化 Oracle 目标数据库。

使用以下优化技术可优化目标：

- 优化平面文件目标。
- 增加数据库检查点时间间隔。
- 使用批量加载。
- 优化 Oracle 目标数据库。

平面文件目标优化

您可以通过优化平面文件目标来提高映射性能。还可以将转换任务推送到命令以提高性能。

要减少平面文件目标瓶颈，请考虑应用以下解决方案：

将转换任务推送到命令而非数据集成服务。

您可以通过将转换任务推送到命令而非数据集成服务来提高映射性能。还可以使用命令来排序或压缩目标数据。在 Developer tool 中，请在平面文件目标的运行时属性中配置命令属性。

在 UNIX 中，请使用任何有效的 UNIX 命令或 Shell 脚本。在 Windows 中，请使用任何有效的 DOS 命令或批处理文件。平面文件写入器将数据发送到命令而非平面文件目标。

例如，请使用以下命令从目标数据生成压缩文件：

```
compress -c - > MyTargetFiles/MyCompressedFile.Z
```

写入服务进程节点的本地平面文件目标。

如果数据集成服务在单个节点上运行并写入平面文件目标，您可以通过写入服务进程节点的本地平面文件目标来优化映射性能。

数据库检查点时间间隔

数据集成服务每次等待数据库执行检查点时，其性能都会降低。

要减少数据库检查点瓶颈，请考虑应用以下解决方案：
增加数据库中的检查点时间间隔。

要减少检查点数量并提高性能，请增加数据库中的检查点时间间隔。

虽然您在减少检查点数量时提高了性能，但是，如果数据库意外关闭，您的恢复时间也将增加。

批量加载

使用批量加载时，数据集成服务将绕过数据库日志，这样可以提高性能。

请考虑应用以下解决方案，以减少批量加载瓶颈：
在数据对象的写入属性中配置批量加载。

您可以使用批量加载来提高用于将大量数据插入到 DB2、Sybase ASE、Oracle 或 Microsoft SQL Server 数据库的映射性能。

如果不写入数据库日志，目标数据库将无法执行回滚。因此，您可能无法执行恢复。使用批量加载时，请权衡提高映射性能的重要性与恢复不完整映射的能力。

数据库目标优化

您可以通过检查存储子句、空间分配以及回滚或撤消段来优化目标数据库。

要减少数据库目标瓶颈，请考虑应用以下解决方案：
验证数据库是否在恰当的表空间中存储回滚或撤消段（最好存储在不同的磁盘上）。

写入数据库时，数据库在加载期间使用回滚或撤消段。要求数据库管理员确保数据库在恰当的表空间中存储回滚或撤消段（最好存储在不同的磁盘上）。回滚或撤消段还应具有恰当的存储子句。

调整数据库恢复日志。

要优化数据库，请调整数据库恢复日志。数据库使用恢复日志记录加载操作。请确保恢复日志大小和缓冲区大小是最优的。对于 Oracle 数据库，您可以查看 init.ora 文件中的恢复日志属性。

通过 IPC 协议连接到 Oracle 数据库。

如果数据集成服务在单个节点上运行，并且 Oracle 实例为服务进程节点的本地实例，您可以通过使用 IPC 协议连接到 Oracle 数据库来优化性能。您可以在 listener.ora 和 tnsnames.ora 中设置 Oracle 数据库连接。

第 3 章

源优化

本章包括以下主题：

- [源优化概览, 15](#)
- [平面文件源优化, 15](#)
- [查询优化, 16](#)
- [条件筛选器, 16](#)
- [选择相异, 17](#)
- [提示, 17](#)
- [约束, 18](#)
- [自定义数据对象优化, 19](#)
- [数据库源优化, 20](#)

源优化概览

要使数据集成服务能够有效地读取源数据，请优化平面文件、关系和自定义数据源。

使用以下优化技术可优化源：

- 有效地读取源数据。
- 使用查询优化技术。
- 使用包含 SQL 查询的条件筛选器。
- 从源中选择唯一值。
- 对 SQL 查询应用提示。
- 配置逻辑数据对象、物理数据对象和虚拟表的约束。
- 配置待优化的自定义数据对象。
- 配置待优化的 Oracle、Sybase 和 Microsoft SQL Server 数据库。

平面文件源优化

要使数据集成服务能够有效地读取源数据，请为平面文件源配置格式属性。

对于平面文件源瓶颈，请考虑应用以下解决方案：

不要在带分隔符的平面文件的格式属性中使用引号或转义符。

如果您指定了转义符，数据集成服务会将分隔符作为字符串中嵌入的常规字符进行读取。如果源文件不包含引号或转义符，您可以略微提高映射性能。

设置数据集成服务每行读取的字节数。

如果映射从平面文件源中读取数据，您可以通过设置数据集成服务每行读取的字节数来提高映射性能。请在平面文件源的运行时属性中配置“换行缓冲区长度”属性。

默认情况下，数据集成服务每行读取 1024 字节。如果源文件中的每个行都少于默认设置，您可以在映射属性中缩短换行缓冲区长度。

查询优化

如果映射联接一个自定义数据对象中的多个源表，您可以通过优化带优化提示的查询来提高性能。此外，添加索引等优化方法可能会使包含 ORDER BY 或 GROUP BY 子句的单个表选择语句受益。

对于查询瓶颈，请考虑应用以下解决方案：

创建优化器提示以告知数据库如何针对一组特定的源表执行查询。

通常情况下，数据库优化器决定处理源数据的最有效方法。但是，您了解的源表相关属性可能数据库优化器不了解。数据库管理员可以创建优化器提示，以告知数据库如何针对一组特定的源表执行查询。

配置优化器提示以尽快开始返回行，而非立即返回所有行。

如果查询开始执行的时间与数据集成服务收到第一行数据的时间之间存在较长的延迟，请使用优化提示。配置优化器提示以尽快开始返回行，而非立即返回所有行。这样将允许数据集成服务在执行查询的同时并行处理行。

针对 ORDER BY 或 GROUP BY 列创建索引。

针对 ORDER BY 或 GROUP BY 列创建索引可能会使包含 ORDER BY 或 GROUP BY 子句的查询受益。优化查询后，请使用 SQL 替代选项以充分利用这些修改。

将数据库配置为运行并行查询。

还可以将源数据库配置为运行并行查询以提高性能。有关配置并行查询的详细信息，请参阅数据库文档。

条件筛选器

由于缺少索引，源数据库中的简单源筛选器有时可能会对性能造成负面影响。您可以在自定义数据对象中使用条件筛选器以提高性能。

对于条件筛选器瓶颈，请考虑应用以下解决方案：

对同时从相同源中读取数据的多个映射使用条件筛选器。

如果多个映射同时从相同源中读取数据，条件筛选器可能会提高性能。

但是，如果您筛选源数据库中的源数据，某些映射执行的速度可能会更快。您可以同时使用数据库筛选器和条件筛选器对映射进行测试，以确定哪种方法能够提高性能。

选择相异

您可以通过“选择相异”选项从自定义数据对象中的源选择唯一值。使用“选择相异”时，数据集成服务将向默认 SQL 查询中添加 SELECT DISTINCT 语句。

对于“选择相异”瓶颈，请考虑应用以下解决方案：
使用“选择相异”选项筛选数据流中不必要的早期数据。

如果您希望数据集成服务从源中选择唯一值，请对自定义数据对象使用“选择相异”选项。使用“选择相异”选项筛选数据流中不必要的早期数据。这样可以提高性能。

例如，您可能会使用“选择相异”选项从列出总销售额的表中提取唯一客户 ID。在映射中使用自定义数据对象时，数据集成服务会筛选掉数据流中不必要的早期数据，这样可以提高性能。

提示

您可以向源 SQL 查询中添加提示，以将说明传递到数据库优化器。优化器使用提示以选择用于访问源的查询运行计划。

“提示”字段在关系数据对象实例或自定义数据对象的查询视图中显示。源数据库必须为 Oracle、Sybase、IBM DB2 或 Microsoft SQL Server。“提示”字段不对其他数据库类型显示。

数据集成服务生成源查询时，您在 Developer 工具中输入 SQL 提示后，数据集成服务会立即将其添加到查询中。数据集成服务不解析这些提示。运行包含源的映射时，映射日志将显示在查询中包含提示的查询。

数据集成服务根据数据库类型在查询中的某个位置插入 SQL 提示。有关提示语法的信息，请参阅数据库文档。

Oracle

数据集成服务直接在 SELECT/UPDATE/INSERT/DELETE 关键字后添加提示。

```
SELECT /*+ <提示> */ FROM ...
```

“+”指示提示的开头。

提示包含在注释中 (/* ... */ 或 --..., 直至行尾)

Sybase

数据集成服务在查询后添加提示。请在提示中配置一个计划名称。

```
SELECT ... PLAN <计划>
```

```
select avg(price) from titles plan "(scalar_agg (i_scan type_price_ix titles ))"
```

IBM DB2

您可以输入 optimize-for 子句作为提示。数据集成服务在查询的结尾添加该子句。

```
SELECT ... OPTIMIZE FOR <n> ROWS
```

optimize-for 子句告知数据库优化器查询可能会处理的行数。该子句不限制行数。如果数据库处理的行数大于 <n>，性能可能会降低。

Microsoft SQL Server

数据集成服务在查询结尾添加提示，作为 OPTION 子句的一部分。

```
SELECT ... OPTION ( <query_hints> )
```

提示规则和准则

为 SQL 查询配置提示时，请遵循以下规则和准则：

- 如果您启用了下推优化，或者如果您在关系数据对象中使用半联接优化，原始源查询可能会发生变化。数据集成服务不对已修改的查询应用提示。
- 您可以将提示与联接和筛选器替代结合，但如果您配置了 SQL 替代，SQL 替代的优先级将更高，并且数据集成服务不应用其他替代。
- **查询**视图显示简单视图或高级视图。如果您输入了包含筛选器的提示，请在简单视图中对替代进行排序或联接，Developer 工具将在高级视图中显示完整查询替代。

创建提示

创建将说明发送到数据库优化器的提示，以确定查询计划。

1. 打开自定义数据对象或关系数据对象实例。
2. 选择**读取**视图。
3. 选择“输出转换”。
4. 选择**查询**属性。
5. 选择简单查询。
6. 单击**提示**字段旁边的**编辑**。
此时将显示**提示**对话框。
7. 在 **SQL 查询**字段中输入提示。
Developer 工具不验证该提示。
8. 单击**确定**。
9. 保存数据对象。

约束

数据集成服务可以从关系源、平面文件源、逻辑数据对象或虚拟表读取约束。约束是数据行上的值必须满足的条件表达式。

当数据集成服务读取约束时，它会依据适用的优化方法删除数据行中计算结果不为 TRUE 的行。

在您设置约束前，必须验证源数据是否满足约束所设置的条件。

例如，源数据库有一个 AGE 列，其中多个行的 AGE 均小于 70。您可以在源数据库上将约束设置为 AGE < 70。这样，数据集成就会使用 AGE < 70 这一约束从源数据库读取记录。如果数据集成服务读取到 AGE >= 70 的记录，则会删除 AGE >= 70 的行。

在数据库中，当您连接到数据库时，可以使用 SQL 命令来设置数据库环境的约束。数据集成服务在每次连接到数据库时运行连接环境 SQL。

使用 Developer tool 设置逻辑数据对象、物理数据对象和虚拟表的约束。当您设置约束时，必须输入对每个数据行计算结果均为 TRUE 的表达式。

配置约束

您可以将约束添加到关系数据对象、平面文件数据对象、自定义数据对象、逻辑数据对象和虚拟表。在添加约束后，您可以编辑或删除约束。

1. 从**对象浏览器**视图中打开包含关系数据对象（作为读取转换添加）的映射。或者，打开平面文件数据对象、自定义数据对象、逻辑数据对象或虚拟表。
 - 要对作为读取转换添加到映射的关系数据对象设置约束，请在映射中选择读取转换。在**属性**视图上，选择**高级**选项卡。
 - 要对平面文件数据对象设置约束，请选择**高级**视图并展开**运行时: 读取**部分。
 - 要设置自定义数据对象的约束，请选择**读取**视图，然后选择源转换的**输出**端口。在**属性**视图上，选择**高级**选项卡。
 - 要设置逻辑数据对象的约束，请选择逻辑数据模型，然后选择逻辑数据对象。在**属性**视图上，选择**高级**选项卡。
 - 要设置虚拟表的约束，请从 SQL 端点选择虚拟表。在**属性**视图上，选择**高级**选项卡。
2. 单击约束的值字段。
此时将显示**约束**对话框。
3. 单击**新建**，打开表达式编辑器。
4. 配置约束逻辑，将表达式函数和列用作参数。
5. 单击**验证**。
6. 单击**确定**。

自定义数据对象优化

您可以配置自定义数据对象以提高性能。您可以优化 SQL 查询、使用条件筛选器以及从自定义数据对象的源中选择相异值。

对于自定义数据对象瓶颈，请考虑应用以下解决方案：
为数据集成服务创建发出特殊 SELECT 语句的自定义查询以读取源数据。

自定义查询替代数据集成服务用于从源中读取数据的默认查询。

数据集成服务读取源数据时将筛选行。

如果您将筛选条件包括在内，数据集成服务将向默认查询中添加 WHERE 子句。

从源中选择相异值。

如果您选择“选择相异”，数据集成服务将向默认 SQL 查询中添加 SELECT DISTINCT 语句。

应用数据库提示。

您可以向源 SQL 查询中添加提示，以将说明传递到数据库优化器。

配置源数据的约束。

如果您在自定义数据对象中配置平面文件和关系表的约束，数据集成服务会删除数据行中计算结果不为 TRUE 的行。

数据库源优化

如果源数据库为 Oracle，您可以通过使用 IPC 协议连接到 Oracle 数据库来优化数据集成服务性能。还可以将临时数据库移至磁盘阵列以提高性能。

对于数据库源瓶颈，请考虑应用以下解决方案：
使用 IPC 协议连接到 Oracle 数据库。

如果数据集成服务在单个节点上运行，并且 Oracle 实例为服务进程节点的本地实例，您可以通过使用 IPC 协议连接到 Oracle 数据库来优化性能。您可以在 listener.ora 和 tnsnames.ora 中设置 Oracle 数据库连接。

将临时数据库和恢复日志移至磁盘阵列或速度更快的驱动器。

联接数据库中的大型表时，可以为缓存位置使用独立磁盘的冗余阵列 (RAID)。此外，还可以向其他磁盘上的主文件组中添加多个文件，以在磁盘之间划分负载。

第 4 章

转换优化

本章包括以下主题：

- [转换优化, 21](#)
- [汇总器转换优化, 21](#)
- [表达式优化, 22](#)
- [Java 转换优化, 24](#)
- [联接器转换优化, 26](#)
- [查找转换优化, 26](#)
- [排序器转换优化, 28](#)
- [SQL 转换优化, 29](#)
- [转换缓存, 30](#)
- [转换错误消除, 31](#)
- [转换的副作用, 31](#)
- [Web 服务使用者转换优化, 31](#)

转换优化

优化转换后，数据集成服务能够有效地处理映射中的转换。

使用以下优化技术可优化转换：

- 配置待优化的转换。
- 消除转换错误。
- 配置转换缓存。

汇总器转换优化

汇总器转换通常会降低性能，因为这些转换在处理数据之前必须对其进行分组。汇总器转换需要额外的内存来保存中间组结果。

对于汇总器转换瓶颈，请考虑应用以下解决方案：

按简单列分组。

按简单列分组时，您可以优化汇总器转换。如有可能，请在用于 GROUP BY 的列中使用数值而不是字符串和日期。请避免在汇总器表达式中使用复杂表达式。

使用已排序输入。

要提高映射性能，请对汇总器转换的数据进行排序。请使用“已排序输入”选项对数据进行排序。

“已排序输入”选项减少了对汇总缓存的使用。使用“已排序输入”选项时，数据集成服务假定所有数据都按组排序。由于数据集成服务读取组的行，因此会执行汇总计算。如有需要，数据集成服务会将组信息存储在内存中。

“已排序输入”选项降低了映射过程中缓存的数据量，提高了性能。使用“已排序输入”选项或排序器转换可将已排序数据传递到汇总器转换。

在使用多个分区的映射中使用“已排序输入”选项时，可以提高性能。

汇总数据之前，请对其进行筛选。

如果您在映射中使用筛选器转换，请在执行汇总器转换之前执行该转换，以减少不必要的汇总。

限制端口连接数。

限制连接的输入/输出端口或输出端口的数量可降低汇总器转换在数据缓存中存储的数据量。

表达式优化

转换中使用的某些表达式可能会降低性能。

对于表达式瓶颈，请考虑应用以下解决方案：

隔离执行速度缓慢的表达式。

执行速度缓慢的表达式会降低映射性能。要隔离执行速度缓慢的表达式，请逐一从映射中删除表达式，然后运行映射以确定运行不包含表达式的映射所需的时间。如果映射运行时间差别较大，请寻找优化执行速度缓慢的表达式的方法。

要评估表达式性能，请完成以下步骤：

1. 确定包含原始表达式的映射的运行时间。
2. 复制该映射并将一半复杂表达式替换为常数。
3. 运行编辑后的映射并确定运行时间。
4. 创建映射的另一个副本，并将另一半复杂表达式替换为常数。
5. 运行编辑后的映射并确定运行时间。

提取通用逻辑。

如果映射在多个位置执行相同的任务，请通过将该任务移至映射中靠前的位置，减少映射执行该任务的次数。例如，您有一个包含五个目标表的映射。每个目标表都需要执行社会保障号查找。请在数据流拆分之前在映射中执行查找转换，而不是执行五次查找。然后将查找结果传递到全部五个目标。

将汇总函数调用次数降至最低。

编写表达式时，请尽量多提取汇总函数调用。每次使用汇总函数调用时，数据集成服务都必须搜索数据并对其进行分析。例如，在以下表达式中，数据集成服务将读取 COLUMN_A，找到总和，然后读取 COLUMN_B，找到总和，最后找到这两个总和的总和：

```
SUM(COLUMN_A) + SUM(COLUMN_B)
```

如果您提取汇总函数调用，数据集成服务会将 COLUMN_A 添加到 COLUMN_B，然后找到这两列的总和，如下所示。

```
SUM(COLUMN_A + COLUMN_B)
```

使用局部变量替换通用表达式。

如果您在一个转换中多次使用同一个表达式，可以将该表达式设为局部变量。可以仅在转换中使用局部变量。但由于只需计算该变量一次，因此会提高性能。

选择数字而非字符串运算符。

与处理字符串操作相比，数据集成服务处理数字操作的速度更快。例如，如果您查找 EMPLOYEE_NAME 和 EMPLOYEE_ID 这两个列中的大量数据，围绕 EMPLOYEE_ID 配置查找可提高性能。

优化 CHAR-CHAR 与 CHAR-VARCHAR 的比较。

数据集成服务在 CHAR 与 VARCHAR 列之间执行比较时，每次查找行中的尾随空格时速度都会减慢。在 Informatica Administrator 中配置数据集成服务以便数据集成服务不裁减 Char 源字段末尾的尾随空格，可以使用 TreatCHARasCHARonRead 选项。

选择 DECODE 而非 LOOKUP。

使用 LOOKUP 函数时，数据集成服务必须查找数据库中的表。使用 DECODE 函数时，您将查找值合并到表达式中，以便数据集成服务不必查找单独的表。因此，要查找一小组未更改的值时，请使用 DECODE 以提高性能。

使用运算符而非函数。

与读取包含函数的表达式相比，数据集成服务读取通过运算符编写的表达式的速度更快。如有可能，请使用运算符编写表达式。例如，您有以下包含嵌套 CONCAT 函数的表达式：

```
CONCAT( CONCAT( CUSTOMERS.FIRST_NAME, ' ' ) CUSTOMERS.LAST_NAME)
```

您可以按如下所示使用 || 运算符重新编写该表达式：

```
CUSTOMERS.FIRST_NAME || ' ' || CUSTOMERS.LAST_NAME
```

优化 IIF 函数。

IIF 函数可以返回值和操作，这样将允许使用更加紧凑的表达式。例如，您有一个包含三个 Y/N 标志的源：FLG_A、FLG_B、FLG_C。您希望根据每个标志的值返回值。

您使用以下表达式：

```
IIF( FLG_A = 'Y' and FLG_B = 'Y' AND FLG_C = 'Y',  
VAL_A + VAL_B + VAL_C,  
IIF( FLG_A = 'Y' and FLG_B = 'Y' AND FLG_C = 'N',  
VAL_A + VAL_B,  
IIF( FLG_A = 'Y' and FLG_B = 'N' AND FLG_C = 'Y',  
VAL_A + VAL_C,  
IIF( FLG_A = 'Y' and FLG_B = 'N' AND FLG_C = 'N',  
VAL_A ,  
IIF( FLG_A = 'N' and FLG_B = 'Y' AND FLG_C = 'Y',  
VAL_B + VAL_C,  
IIF( FLG_A = 'N' and FLG_B = 'Y' AND FLG_C = 'N',  
VAL_B ,  
IIF( FLG_A = 'N' and FLG_B = 'N' AND FLG_C = 'Y',  
VAL_C,  
IIF( FLG_A = 'N' and FLG_B = 'N' AND FLG_C = 'N',  
0.0,  
))))))
```

Java 转换优化

映射中的某些 Java 转换可能会降低性能。

请考虑应用以下解决方案，以提高 Java 转换的性能：

启用通过 Java 转换执行早期选择或推入筛选器优化方法。

您可以在 Java 转换中启用早期选择或推入优化。请在 Java 转换的**优化器接口**选项卡中更新代码段。

通过 Java 转换执行早期选择优化

如果 Java 转换没有副作用，您可以为早期选择优化启用主动或被动 Java 转换。优化器通过 Java 转换传递筛选器逻辑，并根据需要修改筛选条件。

要查看早期选择优化的代码段，请在**优化器接口**选项卡的导航器中选择“PredicatePushOptimization”。

allowPredicatePush

布尔型。启用早期选择。请将此函数更改为返回 true 结果和消息，以启用早期选择。默认值为 false，并且函数将返回一条消息，指出不支持优化。

```
public ResultAndMessage allowPredicatePush(boolean ignoreOrderOfOp) {
    // To Enable PredicatePushOptimization, this function should return true
    //return new ResultAndMessage(true, "");
    return new ResultAndMessage(false, "Predicate Push Optimization Is Not Supported");
}
```

canGenerateOutputFieldEvalError

布尔型。指示 Java 转换是否能够返回输出字段错误，例如除数为零错误。如果 Java 转换不生成输出字段错误，请将函数更改为返回 false。Java 转换能够生成字段错误时，数据集成服务无法使用早期选择优化。

```
public boolean canGenerateOutputFieldEvalError() {
    // If this Java transformation can never generate an output field evaluation error,
    // return false.
    return true;
}
```

getInputExpr

返回一个 Informatica 表达式，用于描述输入字段中的哪些输入值组成输出字段。优化器需要了解哪些输入字段组成输出字段，才能通过转换推送筛选器逻辑。

```
public InfaExpression getInputExpr(TransformationField field,
    TransformationDataInterface group) {
    // This should return an Informatica expression for output fields in terms of input fields
    // We will only push predicate that use fields for which input expressions are defined.
    // For example, if you have two input fields in0 and in1 and three output fields out0, out1, out2
    // out0 is the pass-through of in1, out2 is sum of in1 and in2, and out3 is unknown, the code should
    be:
    //if (field.getName().equals("out0"))
    //    return new InfaExpression("in0", instance);
    //else if (field.getName().equals("out1"))
    //    return new InfaExpression("in0 + in1", instance);
    //else if (field.getName().equals("out2"))
    //    return null;
    //    return null;
}
```

例如，某个映射中包含筛选器表达式 "out0 > 8"。Out0 是指 Java 转换中 out0 输出端口的值。您可以将 out0 的值定义为 in0 输入端口 + 5 的值。优化器能够通过早期选择优化将以下表达式 "(in0 + 5) > 8" 推送到 Java 转换之外。如果输出字段中不包含输入字段表达式，则可以返回空值。优化器不会将筛选器表达式推送到不包含输入表达式的输出字段之外。

您可能会将以下代码包括在内：

```
if (field.getName().equals("out0"))
    return new InfaExpression("in0 + 5", instance);
else if (field.getName().equals("out2"))
    return null;
```

inputGroupsPushPredicateTo

返回能够接收筛选器逻辑的组列表。Java 转换包含一个输入组。请不要为 Java 转换修改此函数。

```
public List<TransformationDataInterface> inputGroupsPushPredicateTo(
    List<TransformationField> fields) {
    // This functions returns a list of input data interfaces to push predicates to.
    // Since JavaTx only has one input data interface, you should not have to modify this function
    AbstractTransformation tx = instance.getTransformation();
    List<DataInterface> dis = tx.getDataInterfaces();
    List<TransformationDataInterface> inputDIs = new ArrayList<TransformationDataInterface>();
    for (DataInterface di : dis){
        TransformationDataInterface tdi = (TransformationDataInterface) di;
        if (tdi.isInput())
            inputDIs.add(tdi);
    }
    if(inputDIs.size() == 1)
        return inputDIs;
    else
        return null;
}
```

通过 Java 转换执行推入优化

如果推入优化的主动 Java 转换没有副作用，并且优化不会影响映射结果，您可以启用此转换。

为 Java 转换配置推入优化时，请为 Java 转换定义存储此转换接收来自优化器的筛选条件的方式。添加用于检查筛选条件的代码。如果 Java 转换能够吸收筛选器逻辑，则会将 true 条件传递回优化器。优化器将从优化的映射中删除筛选器转换。

配置 Java 转换时，您需要编写用于在优化期间将筛选条件存储为转换元数据的代码。还需要编写用于在运行时检索筛选条件以及根据筛选器逻辑删除行的代码。

定义 Java 转换时，请在 Java 转换**优化器接口**选项卡上添加推入优化的代码。要访问推入优化的代码段，请在转换**优化器接口**选项卡的导航器中选择“FilterPushdownOptimization”。

Developer 工具将显示代码段，以启用推入优化以及接收来自优化器的筛选条件。请更新代码段，以启用优化并将筛选器逻辑另存为转换元数据。

isFilterSupported

返回 true 表示启用推入优化。返回 false 表示禁用推入优化。

请将此函数更改为返回 true 以启用推入优化。

```
public ResultAndMessage isFilterSupported() {
    // To enable filter push-into optimization this function should return true
    // return new ResultAndMessage(true, "");
    return new ResultAndMessage(false, "Filter push-into optimization is not supported");
}
```

pushFilter

接收来自优化器的筛选条件。

添加用于检查筛选器以及确定筛选器逻辑是否能够在转换中使用的代码。如果转换能够吸收筛选器，请使用以下方法将筛选条件存储为转换元数据：

storeMetadata(字符串键，字符串数据)

字符串键是指元数据的标识符。您可以将任何字符串定义为字符串键。字符串数据是指您要存储的数据，以便确定运行时要删除的行。例如，字符串数据可以是 Java 转换接收的来自优化器的筛选条件。

```
public ResultAndMessage pushFilter(InfaExpression condition) {  
    // Add code to absorb the filter  
    // If filter is successfully absorbed return new ResultAndMessage(true, ""); and the optimizer  
    // will remove the filter from the mapping  
    // If the filter is not absorbed, return new ResultAndMessage(false, msg);  
    return new ResultAndMessage(false, "Filter push-into optimization is not supported");  
}
```

连接器转换优化

连接器转换会降低性能，因为这些转换在运行时需要额外的空间来保存中间结果。

对于连接器转换瓶颈，请考虑应用以下解决方案：

指定主源作为具有较少重复键值的源。

处理已排序连接器转换时，数据集成服务一次缓存一百个唯一键对应的行。如果主源包含许多具有相同键值的行，则数据集成服务必须缓存多个行，该操作可能会降低性能。

指定主源作为具有较少行的源。

连接器转换将比较详细源和主源的每一行。主源中的行数越少，联接比较发生的迭代数越少，从而加快联接进程的速度。

尽可能在数据库中执行联接。

在数据库中执行联接的速度快于在映射运行期间执行联接的速度。使用的数据库联接类型可能会影响性能。普通联接的速度比外部联接的速度快，且产生的行数较少。有时，无法在数据库中执行联接，例如联接两个不同数据库或平面文件系统中的表。

尽可能联接已排序数据。

配置连接器转换以使用已排序输入。通过最大程度地减少磁盘输入和磁盘输出，数据集成服务可提高性能。如果使用的是大型数据集，将呈现出最显著的性能提高。对于未排序连接器转换，请将具有较少行的源指定为主源。

优化联接条件。

数据集成服务尝试通过读取较小组中的行、查找较大组中的匹配行并执行联接操作来减小一个联接操作的数据集的大小。减小数据集的大小可提高映射性能，因为数据集成服务不再从较大的组源中读取不必要的行。数据集成服务将联接条件移至较大的组源，并仅读取与较小的组匹配的行。

使用半联接优化方法。

在以下情况下，请使用半联接优化方法以提高映射性能：一个输入组具有的行数远大于其他输入组，以及根据联接条件，较大的组具有许多与较小的组不匹配的行。

查找转换优化

查找转换会降低性能，具体取决于查找缓存类型和查找条件。

对于查找转换瓶颈，请考虑应用以下解决方案：

使用最优数据库驱动程序。

数据集成服务可以使用本地数据库驱动程序或 ODBC 驱动程序连接到查找表。本地数据库驱动程序提供的映射性能优于 ODBC 驱动程序。

缓存关系查找或平面文件查找的查找表。

要优化关系源或平面文件源的查找性能，请在转换中启用查找缓存。启用了缓存后，数据集成服务将缓存查找表。运行映射时，数据集成服务将查询查找缓存而非查找表。如果未启用此选项，数据集成服务将逐行查询查找表。

无论您是否缓存查找表，查找查询和处理的结果都相同。但是，使用查找缓存可以提高较小型查找表的映射性能。一般来说，您需要缓存小于 300 MB 的查找表。

缓存逻辑数据对象查找的查找表。

要优化针对逻辑数据对象的查找性能，您可以在数据集成服务上启用数据对象缓存。启用了数据对象缓存后，数据集成服务将缓存逻辑数据对象。要启用数据对象缓存，必须将映射部署到应用程序，启用逻辑数据对象的缓存，然后使用 `infacmd ms runmapping` 命令运行映射。运行映射时，数据集成服务将查询数据对象缓存，而非逻辑数据对象。

如果您从 Developer tool 中运行映射，查找转换将逐行查询逻辑数据对象。

使用恰当的缓存类型。

使用以下类型的缓存可以提高性能：

- 共享缓存。可以在多个转换之间共享查找缓存。可以在相同映射中的转换之间共享未命名缓存。可以在相同或不同映射中的转换之间共享命名缓存。
- 持久性缓存。要保存并重用缓存文件，可以将转换配置为使用持久性缓存。如果您知道查找表在映射两次运行时保持不变，请使用此功能。使用持久性缓存可以提高性能，因为数据集成服务从缓存文件而非从数据库构建内存缓存。

启用并发缓存。

处理包含查找转换的映射时，数据集成服务会在处理已缓存查找转换中的第一行数据时在内存中构建缓存。如果映射中存在多个查找转换，数据集成服务会在查找转换处理第一行数据时按顺序创建缓存。这会减慢查找转换的处理速度。

您可以启用并发缓存以提高性能。附加并发管道数设置为一或更多时，数据集成服务会同时构建缓存，而不是按顺序构建。映射中包含大量可能需要一段时间才能完成的主动转换时（例如汇总器转换、联接器转换或排序器转换），性能会大大提高。启用多个并发管道时，数据集成服务不再等待主动映射完成即开始构建缓存。管道中的其他查找转换也会同时构建缓存。

优化查找条件匹配。

根据查找条件匹配查找缓存数据时，查找转换会对数据进行排序，以确定第一个匹配值和最后一个匹配值。您可以将该转换配置为返回匹配查找条件的任意值。将查找转换配置为返回任意匹配值时，该转换将返回匹配查找条件的第一个值。将查找转换配置为返回第一个匹配值或最后一个匹配值时，该转换不会像平常一样编制所有端口的索引。

使用任意匹配值时，性能会提高，因为该转换不会编制所有端口的索引，这样会降低性能。

减少已缓存的行数。

您可以减少缓存中包含的行数以提高性能。使用“查找 SQL 替代”选项可将 WHERE 子句添加到默认 SQL 语句中。当您将在 WHERE 子句添加到使用动态缓存的查找转换时，请先使用筛选器转换，然后再使用查找转换将行传递到与 WHERE 子句匹配的动态缓存。

替代 ORDER BY 语句。

默认情况下，数据集成服务生成已缓存查找的 ORDER BY 语句。ORDER BY 语句中包含所有查找端口。要提高性能，请禁止默认 ORDER BY 语句，并输入包含较少列的替代 ORDER BY。

数据集成服务始终生成 ORDER BY 语句，即使您在替代中输入了一个语句也是如此。将两个短划线 “--” 置于 ORDER BY 替代之后可禁止生成的 ORDER BY 语句。

例如，某个查找转换使用以下查找条件：

```
ITEM_ID = IN_ITEM_ID  
PRICE <= IN_PRICE
```

该查找转换中包含映射中使用的三个查找端口，即 ITEM_ID、ITEM_NAME 和 PRICE。输入 ORDER BY 语句时，请按照与查找条件中的端口相同的顺序输入列。还必须用引号引起所有数据库保留字。

在查找 SQL 替代中输入以下查找查询：

```
SELECT ITEMS_DIM.ITEM_NAME, ITEMS_DIM.PRICE, ITEMS_DIM.ITEM_ID FROM ITEMS_DIM ORDER BY  
ITEMS_DIM.ITEM_ID, ITEMS_DIM.PRICE --
```

使用具有更大内存的计算机。

要提高映射性能，请在具有大量内存的数据集成服务节点上运行映射。请在不耗尽计算机内存的情况下尽可能提高索引和数据缓存大小。如果数据集成服务节点具有足够的内存，请增大缓存大小，使其能够在内存中保存所有数据，而不需要分页到磁盘。

优化查找条件。

如果您将多个查找条件包括在内，请按以下顺序放置条件以优化查找性能：

- 等于 (=)
- 小于 (<)、大于 (>)、小于或等于 (<=)、大于或等于 (>=)
- 不等于 (!=)

筛选器查找行。

要提高性能，请在构建查找缓存时创建一个筛选条件，以减少从源中检索的查找行数。

编制查找表的索引。

数据集成服务需要查询、排序和比较查找条件列中的值。索引需要包含查找条件中使用的每个列。

您可以提高以下查找类型的性能：

- 已缓存查找。要提高性能，请在查找 ORDER BY 语句中编制列的索引。映射日志文件中包含 ORDER BY 语句。
- 未缓存查找。要提高性能，请在查找条件中编制列的索引。数据集成服务针对传递到查找转换中的每个行发出 SELECT 语句。

优化多个查找。

如果映射中包含多个查找，则即使启用了缓存并且具有足够的堆内存，查找仍会降低性能。请调整查询最大数据量的查找转换以提高整体性能。

如果查找表与映射中的源表位于相同的数据库中，请联接源数据库中的表，而不是使用查找转换。

排序器转换优化

数据集成服务节点上的物理 RAM 未分配足够的内存以对数据进行排序时，排序器转换会降低性能。

对于排序器转换瓶颈，请考虑应用以下解决方案：

分配足够的内存。

要获得最佳性能，请使用小于或等于数据集成服务节点上的可用物理 RAM 量的值配置排序器缓存大小。至少分配 16 MB 物理内存才能使用排序器转换对数据进行排序。默认情况下，排序器缓存大小设置为 16,777,216 字节。如果数据集成服务无法分配足够的内存以对数据进行排序，映射将失败。

如果传入数据量大于排序器缓存大小量，数据集成服务会将数据暂时存储在排序器转换工作目录中。对工作目录中的数据进行排序时，数据集成服务所需的磁盘空间至少为传入数据量的两倍。

SQL 转换优化

数据集成服务每次处理映射中的新查询时，都会调用名为 SQLPrepare 的函数，以创建 SQL 进程并将其传递到数据库。查询针对每个输入行的更改时，可能会降低性能。

对于 SQL 转换瓶颈，请考虑应用以下解决方案：
不在 SQL 转换查询中使用事务语句。

SQL 查询中包含提交和回滚查询语句时，数据集成服务在每次提交或回滚之后必须重新创建 SQL 过程。要优化性能，请不要在 SQL 转换查询中使用事务语句。

启用通过 SQL 转换执行早期选择或推入筛选器优化方法。

要提高性能，请考虑启用通过 SQL 转换执行早期选择或推入优化方法。

通过 SQL 转换执行早期选择优化

如果筛选条件仅引用传递端口，并且 SQL 转换没有副作用，则数据集成服务可以通过 SQL 转换执行早期选择优化。

在下列情况下，SQL 转换会产生负面影响：

- SQL 查询更新新数据库。SQL 查询包含如下语句：CREATE、DROP、INSERT、UPDATE、GRANT 或 REVOKE。
- 转换对未返回任何结果的 SELECT 语句返回空行。这些行可能包含传递端口值、SQL 错误信息或 NumRowsAffected 字段。

启用通过 SQL 转换执行早期选择优化

如果 SQL 转换没有副作用，则可以在 SQL 转换中启用早期选择优化。

1. 在 SQL 转换**高级属性**中启用**仅返回数据库输出**选项。
2. 清除 SQL 转换**高级属性**中的**有副作用**。
3. 如果转换具有 **NumAffectedRows** 端口，请删除该端口。

通过 SQL 转换执行推入优化

通过推入优化，数据集成服务可以将筛选器逻辑从映射中的筛选器转换推送到 SQL 转换中的查询。

启用通过 SQL 转换执行推入优化时，请遵循以下规则和准则：

- 转换 SQL 查询中必须仅包含 SELECT 语句。
- 转换 SQL 查询必须为有效的子查询。
- 筛选条件不能引用 SQL 错误或 NumRowsAffected 字段。

- 输出端口的名称必须与 SQL SELECT 语句中的列名称匹配。在筛选条件中引用输出端口时，数据集成服务会将相应的端口名称推送到 SQL 查询。如果查询中的列与输出端口名称不匹配，可以向 SQL 添加别名。例如 SELECT mycolname1 AS portname1, mycolname2 AS portname2。
- 转换不能有副作用。

通过 SQL 转换执行推入优化示例

SQL 转换按客户 ID 检索订单。在 SQL 转换之后显示的筛选器转换仅返回订购量大于 1000 的行。

数据集成服务将以下筛选器推送到 SQL 转换中的 SELECT 语句：

```
orderAmount > 1000
```

SQL 查询中的每个语句都变为包含该筛选器的 SELECT 语句的独立子查询。

以下查询语句会将原始查询语句显示为 SELECT 语句中的子查询：

```
SELECT <customerID>, <orderAmount>, ... FROM (original query statements) ALIAS WHERE <orderAmount> > 1000
```

如果 SQL 查询具有多个语句，每个语句将包含在一个独立的子查询中。子查询具有相同的语法，包括 WHERE 子句。

端口 *customerID* 和 *orderAmount* 是指 SQL 转换中输出端口的名称。子查询不包含传递端口、SQL 错误或 SQL 统计信息端口。如果您将多个筛选器推送到 SQL 转换，WHERE 子句中 will 包含所有筛选器。

启用通过 SQL 转换执行推入优化

通过在 SQL 转换**高级属性**选项卡中配置属性，可以启用推入优化。

1. 清除**有副作用**。
2. 启用**仅返回数据库输出**。
3. 将**最大输出行数**设置为零。
4. 启用推入优化。

转换缓存

当您运行使用汇总器、联接器、查找、等级或排序器转换的映射时，数据集成服务将在内存中创建缓存来处理转换。如果数据集成服务需要更多空间，它会将溢出值存储在磁盘上的缓存文件中。

对于转换缓存瓶颈问题，请考虑以下解决方案：

配置转换配置来分配足够空间在内存中存储缓存。

要缩短汇总器、联接器、查找、等级或排序器转换的处理时间，请配置转换配置来分配足够空间在内存中存储缓存。当您将缓存内存量配置为等于或大于缓存数据和索引所需的内存量时，可通过降低系统 I/O 开销来提高性能。数据集成服务将缓存文件写入磁盘时，处理时间会因系统 I/O 开销而增加。

默认情况下，数据集成服务在运行时自动配置缓存内存要求。在自动缓存模式下运行映射后，您可以优化转换的缓存大小。通过分析映射日志中的转换统计信息，可确定在内存中处理转换所需的缓存大小。当您将缓存大小配置为使用映射日志中指定的值时，可以确保不会浪费分配的任何内存。但是，最优的缓存大小会因源数据的大小而发生变化。请在后续会话运行后查看会话日志，以监视缓存大小的变化。如果您为可重用的转换配置了某个特定的缓存大小，请验证该缓存大小是否对于映射中的每次转换使用都是最优的。

转换错误消除

如果存在大量转换错误，则会降低数据集成服务的性能。每遇到一个转换错误，数据集成服务就会暂停，以确定错误原因并从数据流中删除导致该错误的行。数据集成服务通常将行写入到数据集成服务日志的映射日志文件中。

对于转换错误瓶颈，请考虑应用以下解决方案：

检查映射日志文件，以了解出现转换错误的位置并评估这些转换约束。

数据集成服务遇到转换错误、映射逻辑冲突以及设置为错误的任何条件（例如空输入）时会出现转换错误。

检查映射日志文件以了解出现转换错误的位置。如果错误与特定转换有关，请评估这些转换约束。

配置较低的跟踪级别。

如果您需要运行生成大量转换错误的映射，可以通过设置一个较低的跟踪级别来提高性能。但是，建议不要长期对转换错误应用此解决方案。

转换的副作用

如果转换返回行并修改对象，或者如果转换与其他对象或函数交互，则该转换有副作用。如果有副作用，则转换可能会修改数据库、增加总量、引发异常、编写电子邮件或调用其他函数。

数据集成服务在优化映射之前会识别有副作用的转换。数据集成服务在无法确定转换是否有副作用时会假定转换有副作用。

数据集成服务能够优化映射时，有副作用的转换会受到限制。如果数据集成服务将映射结果应用到有副作用的转换，则早期选择、分支修剪、全局谓词优化和推入优化会改变映射结果。应用早期选择和推入优化后，筛选器逻辑将从筛选器转换移至尽可能靠近源的位置。如果筛选在副作用函数运行之前进行，映射结果会发生变化。

例如，某个转换收到一个客户 ID 并返回包含订单信息的行。该转换还会将订单写入到文件中。如果数据集成服务在将订单写入文件之前应用筛选器优化，该文件收到的行将少于在映射中较晚进行筛选时收到的行。该转换的副作用是指用于将订单记录写入文件的函数。

以下转换有副作用：

- SQL 转换、Web 服务使用者转换和 Java 转换，除非副作用属性已禁用。
- 可以调用 ABORT() 或 ERROR() 函数、发送电子邮件或调用存储过程的转换。
- 写入文件或数据库的转换。
- 通过变量端口保持计数的转换。例如 COUNT=COUNT+1。

默认情况下，SQL 转换、Web 服务使用者转换和 Java 转换有副作用。如果配置转换以在没有副作用的情况下处理行，可以禁用高级属性中的有副作用属性。如果转换没有副作用，您可以通过在这些转换中配置附加属性来启用优化。

Web 服务使用者转换优化

映射多次调用 Web 服务时，Web 服务使用者转换会降低性能。

对于 Web 服务使用者转换瓶颈，请考虑应用以下解决方案：

将 Web 服务使用者转换配置为使用 cookie 身份验证。

远程 Web 服务服务器会根据 cookie 跟踪 Web 服务使用者用户。某个映射多次调用一个 Web 服务时，可以提高性能。

将 cookie 端口映射到 Web 服务请求消息时，Web 服务提供程序将在响应消息中返回 cookie 值。您可以将该 cookie 值传递到映射中的其他转换下游，或者也可以将该 cookie 值保存在文件中。将 cookie 值保存在文件中时，可以将该 cookie 配置为 Web 服务使用者转换的输入。您可以将 cookie 输出端口传递到任意 Web 服务使用者转换输出组。

启用通过 Web 服务使用者转换执行早期选择或推入筛选器优化方法。

要提高性能，数据集成服务可以应用通过 Web 服务使用者转换执行早期选择或推入优化方法。要应用早期选择优化，Web 服务不能有副作用，并且不能将故障视为错误。要应用推入优化，Web 服务不能有副作用，不能将故障视为错误，并且筛选条件必须引用传递端口。

如果 Web 服务除返回对 Web 服务使用者转换的响应外，还执行其他函数，该服务有副作用。如果 Web 服务修改数据库、写入文件、编写电子邮件、更新计数或调用其他有副作用的 Web 服务，该服务有副作用。

将 Web 服务使用者转换配置为并行发送多个请求。

要提高映射性能，请将 Web 服务使用者转换配置为并行发送多个请求。允许 Web 服务使用者转换创建与 Web 服务的多个并发连接时，可以设置内存消耗限制和并发连接限制数。

通过 Web 服务使用者转换执行早期选择优化

数据集成服务对 Web 服务使用者转换应用早期选择优化方法时，会将映射中 Web 服务使用者转换之前的筛选条件移至靠近源的位置。

启用通过 Web 服务使用者转换执行早期选择优化

如果 Web 服务使用者转换没有副作用，并且不会将故障视为错误，请为该转换启用早期选择优化。

1. 打开 Web 服务使用者转换**高级属性**视图。
2. 清除**将故障视为错误**。
3. 清除**有副作用**。

通过 Web 服务使用者转换执行推入优化

通过推入优化，Web 服务使用者转换在筛选器端口中接收筛选器值。筛选器端口是指未连接的输入端口，您在配置推入优化时将该端口标识为筛选器端口。筛选器端口具有默认值，该值可确保最终用户查询中不包含筛选器时 Web 服务返回所有行。筛选器端口不是传递端口。

注意：筛选器字段必须属于 Web 服务请求中根组的一部分。

筛选器字段不能是传递端口。配置筛选器端口时，端口的默认值更改为筛选条件的值，因而传递输出端口值会发生变化。基于输出传递端口的筛选器返回意外结果。

您可以将多个筛选器表达式推送到 Web 服务使用者转换。每个筛选条件的格式都必须如下所示：

<Field> = <Constant>

筛选条件必须通过 AND 联接。不能通过 OR 联接这些条件。

启用通过 Web 服务使用者转换执行推入优化

如果 Web 服务使用者转换没有副作用，并且不会将故障视为错误，请为该转换启用推入优化。

1. 打开 Web 服务使用者转换**高级属性**视图。
2. 清除**将故障视为错误**。
3. 清除**有副作用**。
4. 单击**推入优化**属性中的**打开**按钮。
5. 在“已优化输入”对话框中选择筛选器端口名称。
您可以选择多个筛选器端口。
6. 单击**输出**列。
7. 对于每个筛选器端口，请选择在 Web 服务响应中包含已筛选列的输出端口。
8. 输入每个筛选器端口的默认值。

注意: 不能为 Web 服务使用者端口配置默认值，除非该端口为筛选器端口。

第 5 章

映射优化

本章包括以下主题：

- [映射优化概览, 34](#)
- [优化方法, 35](#)
- [下推优化, 40](#)
- [一次性读取, 42](#)
- [筛选器优化, 42](#)
- [数据类型转换优化, 42](#)
- [错误跟踪, 43](#)

映射优化概览

要使数据集成服务能够有效地转换和移动数据，请优化映射。映射级别优化可能需要一段时间才能实现，但可以显著提高映射性能。

优化任务适用于常规映射、逻辑数据对象读写映射、虚拟表映射和操作映射。优化目标和源后，请重点执行映射级别优化。

要优化映射，可以执行以下任务：

- 配置包含最少数量的转换和表达式的映射以尽可能完成大量的工作。
- 删除转换之间不必要的链接以将移动的数据量降至最低。
- 选择能够确定数据集成服务可以应用于映射的优化方法的优化器级别。优化映射时，数据集成服务将尝试减少要处理的数据量。例如，数据集成服务可以使用早期选择优化以将筛选器移至更靠近源的位置。可以使用基于成本的优化方法更改联接处理顺序。
- 选择一种下推类型，以使数据集成服务能够确定是否可以将部分或全部转换逻辑下推至源数据库。
- 配置数据对象缓存，以使数据集成服务能够在运行映射时缓存逻辑数据对象并访问预构建的逻辑数据对象。默认情况下，数据集成服务在运行映射时将提取源数据并构建所需的数据对象。数据集成服务能够访问预构建的数据对象时，映射性能将得到提高。
- 配置这些转换时，必须指示 SQL 转换、Web 服务使用者转换和 Java 转换没有副作用。某些转换有限制优化的副作用。例如，如果某个转换写入文件或数据库、添加到计数、引发异常或写入电子邮件，则该转换可能有副作用。在大多数情况下，数据集成服务可以识别有限制优化的副作用的转换。

相关主题：

- [“数据对象缓存” 页面上 51](#)

优化方法

数据集成服务应用优化方法以减少映射中要处理的行数。您可以配置映射的优化器级别以限制数据集成服务应用的优化方法。

数据集成服务可以应用以下优化方法：

- 下推优化
- 早期计划优化
- 早期选择优化
- 分支修剪优化
- 推入优化
- 谓词优化
- 全局谓词优化
- 基于成本的优化
- 船舶数据中心联接优化
- 半联接优化

数据集成服务可以对一个映射同时应用多种优化方法。例如，当您选择普通优化器级别时，数据集成服务会应用早期计划优化、谓词优化、全局谓词优化、分支修剪优化、早期选择优化或推入优化等方法。

优化器级别

数据集成服务根据您配置的优化器级别优化映射。希望映射使用除默认级别外的优化器级别时，请配置优化器级别属性。

您可以选择以下优化器级别之一：

自动

数据集成服务根据执行模式和映射内容应用优化。

无

数据集成服务不应用任何优化。

最小

数据集成服务应用早期计划优化方法。

普通

数据集成服务将应用早期计划、早期选择、分支修剪、推入、全局谓词优化和谓词优化等方法。

完整

数据集成服务将应用基于成本的优化、早期计划、早期选择、分支修剪、谓词、推入、半联接和船舶数据中心联接优化等方法。

默认为“自动”。

从运行菜单或 Developer tool 中的映射编辑器运行映射时，数据集成服务会应用映射配置中的优化器级别。从命令行运行映射时，数据集成服务在应用程序中应用映射部署属性中的优化级别。

注意：数据集成服务不会将下推优化方法与优化器级别一起应用。您可以在映射运行时属性中为映射配置下推优化。

筛选器优化

筛选器优化通过减少传递映射的行数来提高性能。数据集成服务可以应用早期选择优化或推入优化。

数据集成服务应用筛选器优化方法时，会将筛选器移至尽可能靠近映射中的源的位置。如果数据集成服务无法将筛选器移至映射中的转换之前，则可以将筛选器逻辑推入转换中。

早期计划优化方法

数据集成服务应用早期计划优化方法时，将标识未使用的端口并删除这些端口之间的链接。

早期计划优化方法通过减少数据集成服务跨转换移动的数据量来提高性能。数据集成服务处理映射时，会将来自映射中所有已连接端口的数据从一个转换移至另一个转换。在复杂的大型映射中，或者在使用嵌套 Mapplet 的映射中，某些端口可能不向目标提供数据。数据集成服务会标识不向目标提供数据的端口。数据集成服务标识未使用的端口后，将从映射中删除所有未使用的端口之间的链接。

数据集成服务不删除所有链接。例如，数据集成服务不删除以下链接：

- 连接到有副作用的转换的链接。
- 连接到可以调用 ABORT() 或 ERROR() 函数、发送电子邮件或调用存储过程的转换的链接。

如果数据集成服务确定转换中的所有端口都未使用，则将删除指向包含最少数据的端口的链接外的所有转换链接。数据集成服务不会从映射中删除未使用的转换。

Developer 工具默认启用此优化方法。

谓词优化方法

数据集成服务应用谓词优化方法时，将检查映射生成的谓词表达式。数据集成服务将确定其能够简化还是重写表达式以提高映射性能。

数据集成服务运行映射时，将针对映射源生成查询，并根据映射内部的映射逻辑和转换针对查询结果执行操作。查询和操作通常包括谓词表达式。谓词表达式表示数据必须满足的条件。谓词表达式的示例包括筛选器和联接器转换中的筛选器和联接条件。

通过谓词优化方法，数据集成服务还将尝试尽可能早地在映射中应用谓词表达式，以提高映射性能。

数据集成服务根据现有谓词表达式推理关系并创建新的谓词表达式。例如，某个映射中包含具有联接条件“A=B”的联接器转换和具有筛选条件“A>5”的筛选器转换。数据集成服务可以将“B>5”添加到联接条件。

数据集成服务在能够对映射同时应用谓词优化方法和早期选择优化方法时应用这两种方法。例如，数据集成服务通过谓词优化方法创建新的筛选条件时，还会尝试通过早期选择方法将其移至映射中的上游。与单独应用其中任何一种方法相比，同时应用这两种优化方法可以提高映射性能。

如果应用程序提高了性能，数据集成服务将应用谓词优化方法。如果应用程序更改了映射结果或者降低了映射性能，数据集成服务将不应用此方法。数据集成服务默认应用此优化方法。

谓词优化规则和准则

数据集成服务重写谓词表达式时，将对表达式应用数学逻辑以对其进行优化。

数据集成服务可能会执行以下任意或所有操作：

- 跨映射中的谓词表达式标识等效变量，并根据等效性生成简化的表达式。
- 跨映射中的谓词表达式标识冗余谓词并将其删除。
- 从分离子句中提取子表达式，并根据子表达式生成多个简化的表达式。
- 将谓词表达式规范化。
- 尽可能早地在映射中应用谓词表达式。

映射中包含的转换的数据类型在已连接的端口之间不匹配时，数据集成服务可能不对映射应用谓词优化。

满足以下任何条件时，数据集成服务可能不对转换应用谓词优化：

- 转换中包含已连接端口的显式默认值。
- 转换有副作用。
- 转换不允许移动谓词。例如，有副作用的转换可能具有此限制。

Developer 工具默认启用谓词优化方法。

基于成本的优化方法

使用基于成本的优化方法时，数据集成服务将评估映射，生成语义上等价的映射，并运行具有最佳性能的映射。基于成本的优化可缩短执行相邻内部联接操作和完整外部联接操作的映射的运行时间。

语义上等价的映射是指执行相同的功能并生成相同结果的映射。要生成语义上等价的映射，数据集成服务会将原始映射分为多个片段。数据集成服务然后确定其能够优化的映射片段。

优化过程中，数据集成服务可能会添加、删除片段中的转换或者对其进行重新排序。数据集成服务验证已优化片段是否与原始片段生成相同的结果，并生成使用已优化片段的备用映射。

如果数据集成服务确定经过排序的合并联接性能高于嵌套循环联接性能，则它还可以应用经过排序的合并联接。执行联接之前，经过排序的合并联接会使用排序顺序来排列两个数据集。嵌套循环联接则使用嵌套循环来联接两个数据集。数据集成服务可能会使用源中的排序信息，或者如果数据排序的成本低于嵌套循环联接处理的成本，则它可能会创建排序器转换。

数据集成服务生成的全部或大部分映射在语义上都等同于原始映射。数据集成服务使用剖析统计信息或数据库统计信息计算原始映射和每个备用映射的成本。然后标识运行最快的映射。数据集成服务针对最佳备用映射执行验证检查以确保其有效，并生成与原始映射相同的结果。

数据集成服务在内存中缓存最佳备用映射。运行映射时，数据集成服务将检索备用映射并运行该映射，而非运行原始映射。

Developer tool 默认不启用此方法。

船舶数据中心联接优化方法

船舶数据中心联接优化方法会尝试定位与较大数据集相邻的较小数据集，以缩短联接处理时间。当两个表的大小相差很大时，数据集成服务会尝试应用船舶数据中心联接优化方法。

例如，数据集成服务可以应用船舶数据中心联接优化方法，以将包含 10,000 行的主表与包含 1,000,000 行的详情表相联接。数据集成服务将在包含较大的详情表的数据库中创建一个临时的暂存表，以执行船舶数据中心联接。然后，数据集成服务会将较小的主表复制到一个临时表，并将该临时表中的数据与较大的详情表中的数据相联接。数据集成服务执行联接操作后，联接器转换逻辑会在数据库中进行处理。

应用船舶数据中心联接优化方法之前，数据集成服务将执行分析以确定船舶数据中心联接优化是否可行且值得执行。如果分析确定此方法可能会提高性能，数据集成服务将对映射应用此方法。数据集成服务随后将重新分析映射，以确定是否有其他执行船舶数据中心联接优化的机会。如果合适，则会执行其他优化。

Developer tool 默认不启用此方法。

提高性能的船舶数据中心联接要求

船舶数据中心联接优化方法有时无法提高性能。以下因素会影响通过船舶数据中心联接优化实现的映射性能：

- 联接器转换主源具有的行数必须远少于详细信息源。
- 详细信息源必须足够大，这样才有必要进行优化。如果详细信息源不够大，则数据集成服务会发现从主源和详细信息源读取所有数据而不应用船舶数据中心联接优化方法速度更快。

船舶数据中心联接优化规则和准则

如果转换满足以下要求，则数据集成服务可以对联接器转换应用船舶数据中心联接优化：

- 联接类型必须为普通、主外联接或详细外联接。
- 详细信息管道必须源自关系源。
- 如果映射使用基于目标的提交，则联接器转换范围必须为“全部输入”。
- 主管道和详细信息管道不能共享任何转换。
- 映射在详细信息源与联接器转换之间不能包含分支。
- 如果包含联接详细信息端的数据库为不支持 Unicode 编码的 IBM DB2 数据库，数据集成服务将无法应用船舶数据中心联接优化方法。

半联接优化方法

半联接优化方法尝试通过修改映射中的联接操作来减少从源中提取的数据量。

在以下情况下，数据集成服务将对联接器转换应用半联接优化方法：一个输入组具有的行数远大于其他输入组，而且根据联接条件，较大组中的许多行在较小组中都没有匹配行。数据集成服务尝试通过读取较小组中的行、查找较大组中匹配的行并执行联接操作来减小一个联接操作数的数据集的大小。减小数据集的大小可提高映射性能，因为数据集成服务不再从较大的组源中读取不必要的行。数据集成服务将联接条件移至较大的组源，并仅读取与较小的组匹配的行。

应用半联接优化方法之前，数据集成服务将执行分析以确定半联接优化方法是否可行且值得执行。如果分析确定此方法可能会提高性能，数据集成服务将对映射应用此方法。数据集成服务随后分析映射，以确定是否有其他执行半联接优化的机会。如果合适，则会执行其他优化。

Developer tool 默认不启用此方法。

已提高性能半联接优化要求

半联接优化方法有时无法提高性能。以下因素影响通过半联接优化实现的映射性能：

- 联接器转换主源具有的行数必须远少于详细信息源。
- 详细信息源必须足够大，以弥补优化。数据集成服务应用半联接优化时，此方法会将部分开销时间添加到映射处理过程。如果详细信息源非常小，应用半联接方法所需的时间可能会超过处理详细信息源中的所有行所需的时间。
- 数据集成服务必须能够为联接器转换获取源行计数统计信息，以便正确地比较常规联接操作与半联接操作的时间要求。

半联接优化规则和准则

如果转换满足以下要求，数据集成服务可以对接器转换应用半联接优化：

- 联接类型必须为普通、主外联接或详细外联接。联接器转换无法执行完整外联接。
- 详细信息管道必须源自关系源。
- 联接条件必须是有效的“排序-合并-联接”条件。即，每个子句必须是一个主端口和一个详细信息端口的等式。如果存在多个子句，这些子句必须通过 AND 联接。
- 如果映射不使用基于目标的提交，则联接器转换范围必须为“所有输入”。
- 主管道和详细信息管道不能共享任何转换。
- 映射在详细信息源与联接器转换之间不能包含分支。

早期选择优化方法

数据集成服务应用早期选择优化方法时，将拆分、移动或删除映射中的筛选器转换。数据集成服务会将筛选器移至映射上方，更靠近源的位置。

如果筛选条件是一个连词，数据集成服务可能会拆分筛选器转换。例如，数据集成服务可能会将筛选条件“A>100 AND B<50”拆分为两个更简单的条件“A>100”和“B<50”。数据集成服务拆分筛选器时，会将简化的筛选器移至映射管道上方，更靠近源的位置。拆分筛选器时，数据集成服务会分别将筛选器移至管道上方。

在 Developer tool 中选择普通或完全优化器级别时，默认启用早期选择优化方法。使用自动优化器级别时，在以下情况下，会启用早期选择优化方法：

- 映射在本地环境或 Blaze 引擎上运行。
- 映射包含支持将筛选器推送到源的数据源。

如果优化不提高性能，您可以禁用早期选择。

如果筛选器转换前面显示的转换具有副作用，数据集成服务会忽略早期选择优化。数据集成服务无法确定 SQL 转换、Web 服务使用者转换和 Java 转换是否具有副作用。如果这些转换没有副作用，则可为其配置早期选择优化。

全局谓词优化方法

当使用全局谓词优化方法时，数据集成服务可尽早从映射中移除可以筛选掉的行。这样可减少映射需要处理的行数。全局谓词优化方法包括谓词优化方法和早期选择方法。

例如，某个映射中包含具有联接条件“A=B”的联接器转换和具有筛选条件“A>5”的筛选器转换。数据集成服务可以将“B>5”添加到联接条件，并将筛选器转换移至更靠近源的位置。

全局谓词优化方法能够比谓词优化方法更高效地应用谓词表达式。全局谓词优化方法不但会确定是否可以简化或重写表达式以提高映射性能，而且会尝试尽早从映射中应用谓词表达式以提高映射性能。

当映射包含嵌套联接器或每个分支上都具有筛选器的分支时，全局谓词优化方法可推断筛选器并将这些筛选器推送至更靠近源的位置。当使用全局谓词优化方法时，数据集成服务可以拆分筛选器，还可以将筛选器移至更靠近源的位置或将筛选器从映射中移除。

分支修剪优化方法

数据集成服务可以对不向映射中的目标提供任何行的转换应用分支修剪优化方法。

如果筛选条件对数据行的计算结果为 FALSE，则数据集成服务可能会删除筛选器转换。例如，一个映射有两个筛选器转换，可从两个关系源中筛选数据。一个筛选器转换的筛选条件为 Country=US，另一个筛选器转换的筛选

条件为 Country=Canada。联合转换可联接两个关系源，且筛选条件为 Country=US。数据集成服务可能会从映射中删除筛选条件为 Country=Canada 的筛选器转换。

在您选择普通或完整优化器级别时，Developer 工具默认启用分支修剪优化方法。如果通过将优化器级别设置为最低或无仍无法提高优化性能，则您可以禁用分支修剪。

推入优化方法

通过推入优化，数据集成服务可以立即将筛选器转换逻辑移至映射中的筛选器转换的上游转换中。推入优化通过减少传递映射的行数来提高性能。

如果转换有副作用，数据集成服务不会将筛选器逻辑移至其他转换中。数据集成服务无法确定 SQL 转换、Web 服务使用者转换和 Java 转换是否有副作用。但是，您可以为推入优化配置 SQL 转换、Web 服务使用者转换和 Java 转换。

下推优化

数据集成服务应用下推优化时，会将转换逻辑推送到源数据库。数据集成服务会将转换逻辑转换为 SQL 查询，并将 SQL 查询发送至数据库。源数据库运行 SQL 查询以处理转换。

如果源数据库处理转换逻辑的速度能够比数据集成服务快，下推优化将提高映射性能。数据集成服务从源中读取的数据也较少。

数据集成服务推送至源数据库的转换逻辑量取决于数据库、转换逻辑和映射配置。数据集成服务将处理所有无法推送至数据库的转换逻辑。

应用下推优化时，数据集成服务将通过从源到目标或者直到到达无法推送至源数据库的下游转换来分析优化映射。数据集成服务将为每个下推了转换逻辑的源生成并执行 SELECT 查询。如果目标已推送至数据库，则数据集成服务还可以生成 INSERT 查询。数据集成服务将读取 SQL 查询的结果并处理映射中的其余转换。

在映射运行时属性中选择了下推类型后，数据集成服务会将下推优化应用到该映射。

您可以选择下列下推类型：

- 无。此选项表示为映射选择的是“无”下推类型。
- 源。数据集成服务会尝试将尽可能多的转换逻辑下推至源数据库。
- 完整。数据集成服务会将全部转换逻辑推送至源数据库。

还可为下推类型创建字符串参数并使用以下参数值：

- 无
- 源
- 完整

完整下推优化

当应用完整下推优化时，数据集成服务会将映射中的全部转换逻辑推送至源数据库。您可以在映射运行时属性中配置完整下推。

当源和目标位于同一个数据库，或者诸如汇总器转换和筛选器转换等转换在源数据库中进行处理并减少了移动的数据量时，完整下推优化是理想的优化方法。例如，如果映射包含 Teradata 源和 Teradata 目标，则您可以配置完整下推优化，以将 Teradata 源数据库中所有待处理的转换逻辑推送至 Teradata 目标数据库。

如果对包含更新策略转换的映射配置完整下推，则您必须确定该映射的下推兼容性。

在以下情况下，数据集成服务可以下推包含更新策略转换的映射：

- 连接到更新策略转换的目标转换接收多个没有相同键的行。
- 连接到更新策略转换的目标转换接收多个具有相同键且可重新排序的行。

在以下情况下，数据集成服务无法下推包含更新策略转换的映射：

- 连接到更新策略转换的目标转换接收多个具有相同键但不可重新排序的行。

您还可以在映射中使用下推兼容性参数。可以使用以下参数值：

- noMultipleRowsWithSameKeyOnTarget
- reorderAllowedForMultipleRowsWithSameKey
- reorderNotAllowedForRowsWithSameKey

数据集成服务可对以下源使用完整下推优化：

- Amazon Redshift
- Greenplum
- IBM DB2
- Microsoft SQL Server
- Netezza
- Oracle
- SAP HANA
- Snowflake
- Teradata

源下推

当应用源下推时，数据集成服务会分析源到目标的映射，在到达下游转换之前，它无法推送至源数据库。

数据集成服务将基于可以推送至数据库的每个转换的转换逻辑来生成和执行 SELECT 语句。然后，会读取此 SQL 查询的结果并处理剩余转换。

如果源和目标驻留在不同的数据库中，您可以将映射配置为使用源下推。例如，如果映射包含 Teradata 源和 Oracle 目标，则您可以配置源下推，以将部分待处理的转换逻辑推送至 Teradata 源。

下推优化规则和准则

数据集成服务可以将转换逻辑推送到源数据库。

以下规则和准则适用于下推优化：

- 如果源位于同一个数据库管理系统中，并且使用相同的连接，则数据集成服务可以将查找转换逻辑和联接器转换逻辑推送到源数据库。
- 数据集成服务无法将转换逻辑推送到具有二进制数据类型的源。
- 当您拥有 IBM DB2 数据源且列精度小数数据类型在 28 到 31 位之间时，数据集成服务禁用下推优化。
- 数据集成服务默认启用针对 SQL 数据服务或 Web 服务的下推优化。您无法禁用针对 SQL 数据服务或 Web 服务的下推优化。
- 数据集成服务无法推送特定的汇总器转换，这种汇总器转换包含的表达式在未分组端口具有汇总和非汇总函数。

一次性读取

一次性读取允许您使用一个自定义数据对象填充多个目标。如果您的多个映射使用相同的资源，请考虑使用一次性读取。

对于一次性读取瓶颈，请考虑应用以下解决方案：

在一个映射中组合每个映射的转换逻辑，并为每个源使用一个自定义数据对象。

数据集成服务读取每个源一次，然后将数据发送到单独的管道中。某个特定的行可以被所有管道或任意管道组合使用，也可以不被任何管道使用。

例如，您有“采购”源表，并且每天使用该源执行汇总和评级。如果您将汇总器和等级转换置于独立的映射中，则可以强制数据集成服务读取相同的源表两次。但是，如果您在一个包含一个源限定符的映射中包括汇总和评级逻辑，数据集成服务将读取“采购”源表一次，然后将相应的数据发送到独立的管道。

从映射中提取公用函数。

将映射更改为利用一次性读取时，您可以通过从映射中提取公用函数来优化此功能。例如，如果您需要从汇总器和等级转换的“价格”端口中减去一个百分比，则可以在拆分管道之前减去该百分比，以将工作量降至最低。您可以使用表达式转换减去该百分比，并在转换后拆分映射。

筛选器优化

您可以通过在自定义数据对象中执行筛选以及将筛选置于映射中靠前的位置来优化映射。

对于筛选器瓶颈，请考虑应用以下解决方案：

在自定义数据对象中使用筛选器以在源中删除行。

如果您从映射中筛选行，则可以在数据流中尽早执行筛选以提高效率。在自定义数据对象中使用筛选器以在源中删除行。自定义数据对象限制从关系源中提取的行集。

如果不能在自定义数据对象中使用筛选器，请使用筛选器转换并将其移至尽可能靠近该自定义数据对象的位置，以尽可能在数据流中删除不必要的数据。筛选器转换限制发送到目标的行集。

在不希望保留被拒绝的行时，在更新策略转换中使用筛选器。

要提高映射性能，您还可以在不希望保留被拒绝的行时，使用筛选器转换以从更新策略转换中删除被拒绝的行。

避免在筛选条件中使用复杂表达式。

请避免在筛选条件中使用复杂表达式。要优化筛选器转换，请在筛选条件中使用简单的整数或 true/false 表达式。

筛选器转换筛选映射中的数据。筛选器转换从任意类型的源中筛选行。自定义数据对象从关系源中筛选行。筛选器转换从任意类型的源中筛选行。

数据类型转换优化

您可以通过消除不必要的数据类型转换来提高性能。例如，如果某个映射将数据从“整数”列移至“小数”列，然后移回“整数”列，则不必要的数据类型转换会降低性能。请尽可能从映射中消除不必要的数据类型转换。

对于数据类型转换瓶颈，请考虑应用以下解决方案：

在使用查找和筛选器转换执行比较时，使用整数值来代替其他数据类型。

例如，许多数据库都将美国邮政编码信息存储为 Char 或 Varchar 数据类型。如果您将邮政编码数据转换为“整数”数据类型，则查找数据库会将邮政编码 94303-1234 存储为 943031234。这样可帮助提高基于邮政编码的查找比较的速度。

通过端口到端口转换将源日期转换为字符串以提高映射性能。

您可以将目标中的端口保留为字符串或者将端口更改为日期/时间端口。

错误跟踪

要提高性能，请减少数据集成服务在运行映射时生成的日志事件数量。通过映射配置或映射部署属性更新映射优化器级别可以提高映射性能。使用基于成本的优化方法可优化映射。

对于错误跟踪瓶颈，请考虑应用以下解决方案：

将映射属性中的跟踪级别设置为“简洁”

如果映射中包含大量转换错误，而您不需要进行更正，请将映射属性中的跟踪级别设置为“简洁”。数据集成服务在此跟踪级别不写入拒绝数据的错误消息或行级信息。

如果您需要调试映射，并且将跟踪级别设置为“详细”，则可能会在运行映射时遇到严重的性能降级。在调整性能时请不要使用“详细”跟踪级别。映射跟踪级别将替代映射中任何转换特定的跟踪级别。建议不要长期对高级别转换错误应用此解决方案。

更改映射的优化器级别。

如果映射运行时所需的时间过长，您可能希望更改映射的优化器级别。优化器级别决定数据集成服务在运行时对映射应用的优化方法。

请在映射配置或映射部署属性中设置映射的优化器级别。数据集成服务对映射应用不同的优化器级别，具体取决于运行映射的方式。

使用基于成本的优化方法。

使用基于成本的优化方法时，数据集成服务将评估映射、生成语义上等价的映射以及运行具有最佳性能的映射。对于包含多个联接器转换的映射，此方法的效率最高。此方法缩短了执行相邻的未排序内部联接操作的映射的运行时间。

语义上等价的映射是指执行相同的功能并生成相同结果的映射。要生成语义上等价的映射，数据集成服务会将原始映射分为多个片段。数据集成服务然后确定其能够优化的映射片段。

第 6 章

已分区映射优化

本章包括以下主题：

- [已分区映射优化概述, 44](#)
- [使用多个 CPU, 44](#)
- [增加最大并行数值, 45](#)
- [针对分区优化平面文件, 45](#)
- [优化分区的关系数据库, 46](#)
- [针对分区优化转换, 47](#)

已分区映射优化概述

如果具有分区选项，则可以在数据集成服务运行映射时将其启用以实现并行最大化。实现并行最大化后，数据集成服务可将基本数据动态分为多个分区并同时处理所有分区。

如果映射处理大数据集或者包含执行复杂计算的转换，则映射可能要花费较长时间进行处理并导致数据吞吐量降低。为这些映射启用分区时，数据集成服务将使用额外线程来处理映射。

您可以通过执行以下任务来优化已分区映射的性能：

- 在运行映射的节点上使用多个 CPU。
- 增加数据集成服务的最大并行数值。
- 配置针对平面文件数据对象的属性。
- 配置关系数据库以优化分区。
- 配置针对转换的属性。

使用多个 CPU

增加处理线程数会增加运行映射的节点的负载。如果节点包含足够的 CPU 带宽，则并发处理映射中的数据行可提高映射性能。

数据集成服务可以使用多个 CPU 处理包含多个分区的映射。服务可使用的 CPU 个数取决于如下因素：分区点个数、为每个管道阶段创建的线程数以及处理映射所需的资源数量。在两个分区中运行的简单映射速度更快，但其所需的 CPU 数量通常是在单个分区中运行的映射的两倍。

增加最大并行数值

最大并行数确定了处理单个管道阶段的最大并行线程数。您可以基于可用的硬件资源配置数据集成服务的**最大并行数**属性。如果增加最大并行数值，可以减少处理时间。

增加最大并行数值时，请考虑以下准则：

根据可用 CPU 数量增加该值。

可以根据运行映射的节点上可用的 CPU 数量增加最大并行数值。增加最大并行数值后，数据集成服务会使用更多线程来运行映射，而且会利用更多 CPU。在两个分区中运行的简单映射速度更快，但其所需的 CPU 数量通常是在单个分区中运行的映射的两倍。

考虑处理线程总数。

在设置最大并行数值时，请考虑处理线程总数。如果复杂映射导致多个额外分区点，则数据集成服务可能会使用超过 CPU 处理能力的处理线程数。

处理线程总数等于最大并行数值。

请考虑数据集成服务必须运行的其他作业。

如果将最大并行数配置为使每个映射使用大量线程，则可供数据集成服务用于运行其他作业的线程将减少。

或者，更改映射的值。

默认情况下，每个映射的最大并行数设置为“自动”。每个映射都使用为数据集成服务定义的最大并行值。

在 Developer tool 中，开发人员可以在映射运行时属性中更改最大并行数值，以定义特定映射的最大值。当数据集成服务和映射的最大并行数设置为不同的整数值时，数据集成服务将使用两者之中的较小值。

针对分区优化平面文件

启用分区的映射从平面文件源读取或写入平面文件目标时，数据集成服务可以使用多个线程读取平面文件或写入到平面文件。

针对分区优化平面文件源

要在使用多个线程从平面文件读取时实现最佳性能，请将平面文件数据对象配置为优化吞吐量，而不要配置为保留行顺序。

要减少已分区平面文件源的瓶颈，请考虑以下解决方案：

将平面文件数据对象的并发读取分区配置为优化吞吐量。

在平面文件数据对象的高级属性中，将**并发读取分区**属性设置为优化吞吐量。优化吞吐量时，数据集成服务不会保留行顺序，因为数据集成服务不按顺序读取文件或文件列表中的行。

针对分区优化平面文件目标

要在使用多个线程写入到平面文件时实现最佳性能，请将分区配置为将目标输出写入到单独的文件，并配置多个目标目录。

要减少已分区平面文件目标的瓶颈，请考虑以下解决方案：

将分区配置为将目标输出写入到单独的文件。

在平面文件数据对象的高级属性中，将**合并类型**属性设置为**不合并**。数据集成服务会并发针对每个分区将目标输出写入一个单独的文件。如果您需要已合并的目标数据，则“并发合并”类型可比“连续合并”类型更有效地优化性能。

配置多个目标目录。

当多个线程向单个目录写入数据时，映射可能会因输入/输出 (I/O) 争用而遇到瓶颈问题。当多个线程同时向文件系统写入数据时，将出现 I/O 争用。当您配置了多个目录时，数据集成服务会以循环方式确定每个线程的输出目录。

在平面文件数据对象的高级属性中配置输出文件目录。如果管理员在 Administrator 工具中为数据集成服务的**目标目录**属性输入了多个以分号分隔的目录，则使用默认的 TargetDir 系统参数值。或者，您也可以输入其他值来配置多个特定于平面文件数据对象的输出文件目录。

优化分区的关系数据库

启用分区的映射从 IBM DB2 for LUW 或 Oracle 关系数据库进行读写操作时，数据集成服务可以使用多个线程读取关系源或写入到关系目标。

要在使用多个线程从 DB2 for LUW 或 Oracle 关系数据库读取数据或向其中写入数据时优化性能，您可以对源表和目标表进行分区。

注意：如果映射对 DB2 for LUW 或 Oracle 之外的关系数据库进行读写操作，则数据集成服务使用一个读取器线程或一个编写器线程。

优化源数据库以进行分区

要在使用多个线程从 DB2 for LUW 或 Oracle 源数据库读取时实现最佳性能，请验证源表是否已分区并配置为接受并行查询。

要优化源数据库以进行分区，请执行以下任务：

将数据库分区添加到源。

将数据库分区添加到关系源，以增加读取源的数据集成服务查询的速度。如果源没有数据库分区，则数据集成服务会使用一个线程来从源读取数据。

启用并行查询。

关系数据库可能具有支持并行查询数据库的选项。请参阅数据库文档以了解这些选项。如果未启用这些选项，则数据集成服务将连续运行多个分区 SELECT 语句。

将数据分到不同的表空间中。

每个数据库提供了一个用于将数据分到不同表空间的选项。每个表空间可引用唯一的文件系统，这会阻止分区之间的任何 I/O 争用。

增加数据库允许的最大会话数。

数据集成服务会为每个分区创建一个与源数据库的单独连接。增加允许会话的最大数目，以便数据库可以处理大量并发连接。

优化目标数据库以进行分区

要在使用多个线程写入到 DB2 for LUW 或 Oracle 目标数据库时实现最佳性能，请验证目标表是否已分区并配置为并行插入行。

要优化目标数据库以进行分区，请执行以下任务：

将数据库分区添加到 DB2 for LUW 目标。

数据集成服务可以使用多个线程写入不包含数据库分区的 DB2 for LUW 目标。但是，当目标具有数据库分区时，您可以优化加载性能。在这种情况下，每个写入器线程均连接至包含数据库分区的 DB2 for LUW 节点。由于写入器线程均连接至不同的 DB2 for LUW 节点，而不是所有线程都连接至一个主节点，所以提高了性能。

启用并行插入。

关系数据库可能具有支持并行插入到数据库的选项。请参阅数据库文档以了解这些选项。例如，设置 Oracle 数据库中的 `db_writer_processes` 选项和 DB2 for LUW 数据库中的 `max_agents` 选项，以启用并行插入。

将数据分到不同的表空间中。

每个数据库提供了一个用于将数据分到不同表空间的选项。每个表空间可引用唯一的文件系统，这会阻止分区之间的任何 I/O 争用。

增加数据库允许的最大会话数。

数据集成服务会为每个分区创建一个与目标数据库的单独连接。增加允许会话的最大数目，以便数据库可以处理大量并发连接。

设置选项以增强数据库的可扩展性。

关系数据库可能具有用于增强可扩展性的选项。例如，在 Oracle 数据库中禁用存档日志记录和计时统计信息以增强可扩展性。

针对分区优化转换

当通过多个线程运行汇总器转换、联接器转换、等级转换或排序器转换时，数据集成服务会使用缓存分区在这些线程之间划分缓存大小。要针对缓存分区优化性能，请配置多个缓存目录。

注意：查找转换只能使用单个缓存目录。

要减少已分区的汇总器转换、联接器转换、等级转换和排序器转换的瓶颈，请考虑以下解决方案：

配置多个缓存目录。

缓存分区功能可为每个处理汇总器转换、联接器转换、等级转换或排序器转换的分区创建单独的缓存。在缓存分区过程中，每个分区在单独的缓存中存储不同的数据。每个存储包含特定分区所需的行。由于每个线程可并行查询单独的缓存，因此缓存分区功能可优化映射性能。

如果缓存大小小于运行转换所需的内存量，则转换线程会写入到缓存目录，以将溢出值存储在缓存文件中。当多个线程向一个目录写入数据时，映射可能会因 I/O 争用而遇到瓶颈问题。当多个线程同时向文件系统写入数据时，将出现 I/O 争用。当您配置了多个缓存目录时，数据集成服务将以循环方式确定每个转换线程的缓存目录。

对于汇总器转换、联接器转换或等级转换，您可以在**缓存目录**高级属性中配置缓存目录。如果管理员在 Administrator 工具中为数据集成服务的**缓存目录**属性输入了多个以分号分隔的目录，则使用默认的 CacheDir 系统参数值。或者，您也可以输入其他值来配置多个特定于转换的缓存目录。

对于排序器转换，您可以在**工作目录**高级属性中配置缓存目录。如果管理员在 Administrator 工具中为数据集成服务的**临时目录**属性输入了多个以分号分隔的目录，则使用默认的 TempDir 系统参数值。或者，您也可以输入其他值来配置多个特定于转换的缓存目录。

第 7 章

运行时优化

本章包括以下主题：

- [运行时优化概览, 49](#)
- [监视统计信息, 49](#)
- [内存分配, 50](#)
- [数据对象缓存, 51](#)
- [系统优化, 53](#)

运行时优化概览

要优化映射性能，请启用性能功能并调整数据集成服务属性。

根据您的要求在 Administrator 工具中使用以下优化技术可获得最佳性能结果：

- 优化应用程序服务进程。
- 配置监视统计信息以监视系统瓶颈。
- 分配内存以实现最佳系统性能。
- 配置数据对象缓存。
- 优化系统以避免系统延迟和减慢磁盘访问速度。

监视统计信息

监视是服务管理器执行的一项域功能。服务管理器在模型存储库中存储监视配置。使用 Administrator 工具中的“监视”选项卡可监视系统瓶颈，例如选定服务上运行的正在运行、失败、已取消和已完成的作业总数。

对于监视统计信息瓶颈，请考虑应用以下解决方案：

配置域以设置监视。

设置监视时，数据集成服务在模型存储库中存储保留的统计信息和监视报告。保留的统计信息是指与以前运行的集成对象有关的历史信息。监视报告显示与集成对象有关的关键度量。

为域配置监视设置，以指定用于存储有关部署到数据集成服务的对象的运行时统计信息的模型存储库。监视设置应用到域中的所有数据集成服务，并且可能会影响服务性能。

下表介绍了可能会影响服务性能的监视设置：

选项	说明
保留摘要历史数据	模型存储库保存平均数据的天数。如果禁用清除功能，则模型存储库会无限期地保存数据。 默认值为 180。最小值为 0。最大值为 366。
保留详细历史数据	模型存储库保存每分钟数据的天数。如果禁用清除功能，则模型存储库会无限期地保存数据。 默认值为 14。最小值为 1。最大值为 14。
统计信息清除间隔	以天数计的间隔，模型存储库服务将根据此间隔清除天数大于 保留历史数据 选项中所配置值的数据。 默认值为 1 天。
天，时间为	模型存储库服务清除统计信息时的时间。默认值为凌晨 1:00。
可排序记录数上限	可以在 监视 选项卡中排序的记录数上限。如果 监视 选项卡中的记录数大于此值，则您只能按 开始时间 和 结束时间 进行排序。默认值为 3,000。
更新通知的最大延迟时间	数据集成服务将统计信息存储到模型存储库中并在 监视 选项卡中显示统计信息之前缓冲统计信息的最长时间（以秒为单位）。如果数据集成服务在将统计信息存储在模型存储库中之前意外关闭，则统计信息将会丢失。默认值为 10。
显示毫秒数	包括 监视 选项卡中日期和时间字段的毫秒数。

内存分配

要优化映射性能，请在 Administrator 工具中配置数据集成服务的内存属性。

下表介绍了映射服务模块的“每个请求的内存上限”属性：

属性	说明
每个请求的内存上限	<p>“每个请求的内存上限”属性的行为取决于以下数据集成服务配置：</p> <ul style="list-style-type: none">- 服务在单独的本地或远程进程中运行作业，或者服务属性“内存大小上限”的值为 0（默认值）。 在这种情况下，每个请求的内存上限是数据集成服务可以分配给一个请求中所有使用自动缓存模式的转换的最大内存量（字节数）。服务单独为具有特定缓存大小的转换分配缓存。请求所用的总内存可以超过“每个请求的内存上限”值。- 服务在数据集成服务进程中运行作业，并且服务属性“内存大小上限”的值大于 0。 在这种情况下，每个请求的内存上限是数据集成服务可以分配给单个请求的最大内存量（字节数）。请求所用的总内存不可超过“每个请求的内存上限”值。 <p>默认值为 536,870,912。</p>

下表介绍了数据集成服务的执行选项：

属性	说明
内存大小上限	<p>当服务在数据集成服务进程中运行作业时，数据集成服务可以分配用于并行运行所有请求的最大内存量（以字节为单位）。当数据集成服务在单独的本地或远程进程中运行作业时，服务会忽略此值。如果不想限制数据集成服务可以分配的内存量，请将此属性设置为 0。</p> <p>如果该值大于 0，数据集成服务将使用该属性计算并行运行所有请求所允许使用的最大总内存。数据集成服务将按如下所示计算最大总内存：</p> <p>内存大小上限 + 堆大小上限 + 加载程序组件时所需的内存</p> <p>默认值为 0。</p> <p>注意：如果您运行配置文件或数据质量映射，请将此属性设置为 0。</p>

下表介绍了数据集成服务进程的“堆大小上限”属性：

属性	说明
堆大小上限	<p>分配给运行数据集成服务的 Java 虚拟机 (JVM) 的 RAM 量。使用此属性可提高性能。将以下其中一个字母附加到值可指定单位：</p> <ul style="list-style-type: none">- b 代表字节。- k 代表千字节。- m 代表兆字节。- g 代表千兆字节。 <p>默认值为 1024 MB。</p> <p>注意：当数据集成服务需要处理大量数据时，考虑增大堆大小。</p>

数据对象缓存

数据集成服务会使用数据对象缓存来访问预构建的逻辑数据对象和虚拟表。启用数据对象缓存可提高包含逻辑数据对象和虚拟表的映射、SQL 数据服务查询和 Web 服务请求的性能。

默认情况下，数据集成服务在运行映射、SQL 数据服务查询或 Web 服务请求时会提取源数据并构建所需的数据对象。启用数据对象缓存时，数据集成服务可以使用已缓存的逻辑数据对象和虚拟表。

执行以下步骤为应用程序中的逻辑数据对象和虚拟表配置数据对象缓存：

1. 在数据集成服务的缓存属性中配置数据对象缓存数据库连接。
2. 在应用程序中的逻辑数据对象或虚拟表的属性中启用缓存。

默认情况下，数据集成服务的数据对象缓存管理器组件在数据对象缓存数据库中管理逻辑数据对象和虚拟表的缓存表。数据对象缓存管理器管理缓存时，会在每次刷新时将所有数据插入缓存表。如果您想以增量方式更新缓存表，可以选择使用数据库客户端或其他外部工具自己管理缓存表。启用数据对象缓存后，您可以将逻辑数据对象或虚拟表配置为使用用户管理的缓存表。

要使用时区数据类型的时间戳并为 IBM DB2 或 Microsoft SQL Server 启用数据对象缓存，请将部署映射的日期时间格式设置为“YYYY-MM-DD HH24:MI:SS”格式。数据集成服务写入的数据精确到秒。

缓存表的数据类型

数据对象缓存管理器会使用数据集成服务所需的数据类型创建缓存表。如果您使用用户管理的缓存表，请确认缓存表使用数据集成服务需要的数据类型。

虚拟表缓存数据类型

下表列出了虚拟表的缓存表数据类型：

虚拟表数据类型	IBM DB2	Microsoft SQL Server	Oracle
Char	Vargraphic Dbclob, 适用于大于 32672 的精度	Nvarchar Ntext, 适用于大于 4000 的精度	Nvarchar2 Nclob, 适用于大于 2000 的精度
Bigint	Bigint	Bigint	Number
Boolean	Integer	Int	Number
Date	时间戳	Datetime2	时间戳
Double	Double	浮点型	时间戳
Decimal	Decimal	Decimal	Number
Int	Integer	Int	Number
时间	时间戳	Datetime2	时间戳
时间戳	时间戳	Datetime2	时间戳
Varbinary	Blob	Binary 图像, 适用于大于 8000 的精度	原始 Blob, 适用于大于 2000 的精度
Varchar	Vargraphic Dbclob, 适用于大于 32672 的精度	Nvarchar Ntext, 适用于大于 4000 的精度	Nvarchar2 Nclob, 适用于大于 2000 的精度

逻辑数据对象缓存数据类型

下表列出了逻辑数据对象的缓存表数据类型：

逻辑数据对象数据类型	DB2	Microsoft SQL Server	Oracle
Bigint	Bigint	Bigint	Number
Binary	Blob	Binary 图像, 适用于大于 8000 的精度	原始 Blob, 适用于大于 2000 的精度
Date/time	时间戳	Datetime2	时间戳
Double	Double	浮点型	Number
Decimal	Decimal	Decimal	Number
Integer	Integer	Int	Number

逻辑数据对象数据类型	DB2	Microsoft SQL Server	Oracle
String	Vargraphic Dbclob, 适用于大于 32672 的精度	Nvarchar Ntext, 适用于大于 4000 的精度	Nvarchar2 Nclob, 适用于大于 2000 的精度
Text	Vargraphic Dbclob, 适用于大于 32672 的精度	Nvarchar Ntext, 适用于大于 4000 的精度	Nvarchar2 Nclob, 适用于大于 2000 的精度

数据对象缓存优化

缓存性能取决于缓存数据库的性能以及映射、SQL 数据服务和 Web 服务中的对象配置。

请考虑应用以下解决方案以提高缓存性能：

优化缓存数据库。

缓存是否能够实现最佳性能，取决于缓存数据库的速度和性能以及缓存大小。请在缓存数据库内部配置缓存大小。

由于数据对象缓存管理器必须维护刷新操作的旧缓存，因此，缓存必须足够大，以存储两组数据。请使用以下公式估算所需的缓存大小下限：

$$2 * \text{average data object size} * \text{number of data objects}$$

缓存表是只读的。最终用户无法通过 SQL 命令更新缓存表。

为逻辑数据对象定义主键和外键。

数据集成服务生成带有键的逻辑数据对象的缓存时，将创建索引。索引可以提高针对缓存数据库的查询性能。

缓存在映射中联接的逻辑数据对象。

联接已缓存逻辑数据对象时，数据集成服务可以将联接器转换逻辑下推到缓存数据库，即使源数据源自不同的数据库时也是如此。

系统优化

对于系统优化瓶颈，请考虑应用以下解决方案：

提高网络速度。

网络连接速度缓慢可能会降低映射性能。请系统管理员确定网络是否以最优速度运行。减少数据集成服务进程与数据库之间的网络跃点数。

使用多个 CPU。

您可以使用多个 CPU 以并行运行多个映射。

减少分页。

操作系统物理内存不足时，将开始分页到磁盘以释放物理内存。请配置数据集成服务进程计算机的物理内存以将分页到磁盘降至最低。

使用处理器绑定。

在多处理器 UNIX 环境中，数据集成服务可能会使用大量系统资源。使用处理器绑定可控制集成服务进程的处理器使用率。此外，如果源和目标数据库位于相同的计算机上，请使用处理器绑定以限制数据库使用的资源。

第 8 章

SQL 数据服务优化

本章包括以下主题：

- [SQL 数据服务优化概览, 55](#)
- [第三方客户端工具优化, 55](#)
- [SQL 数据服务优化器级别, 56](#)
- [SQL 数据服务的内存和并发请求属性, 59](#)
- [SQL 数据服务的结果集缓存, 60](#)
- [在临时表中保留虚拟数据, 61](#)

SQL 数据服务优化概览

最终用户使用第三方客户端工具针对 SQL 数据服务运行 SQL 查询时，您可以优化该服务以提高性能。如果 SQL 数据服务使用虚拟表映射，您可以优化源、转换和映射。

使用以下优化技术可优化 SQL 数据服务：

- 优化第三方客户端工具。
- 配置 SQL 数据服务优化器级别。
- 为数据集成进程配置并发请求和内存的 SQL 数据服务属性。
- 为 SQL 数据服务配置数据对象缓存。
- 为 SQL 数据服务配置结果集缓存。
- 为 SQL 数据服务中的虚拟表配置约束。

相关主题：

- [“数据对象缓存” 页面上 51](#)

第三方客户端工具优化

处理 SQL 查询并针对 SQL 数据服务运行这些查询时，第三方客户端工具可能会影响性能。优化最终用户可用于针对 SQL 数据服务运行 SQL 查询的第三方客户端工具。

对于第三方客户端工具瓶颈，请考虑应用以下解决方案：

将大型查询结果发送到磁盘上的文件。

如果第三方客户端工具在控制台窗口中显示大型查询结果，则可能会影响性能。

配置第三方客户端工具以禁用加密。

如果第三方客户端工具在提取或显示查询结果时加密数据，则可能会影响性能。

配置第三方客户端工具以预先提取一组行。

如果第三方客户端工具一次提取单个行，则可能会影响性能。

配置第三方客户端工具以禁用在表首次加载时从中读取内容的选项。

如果 BLOB 和 CLOB 数据类型的数据类型设置配置为在表首次加载时从中读取内容，或者如果 BLOB 和 CLOB 数据类型未在查询中使用，则第三方客户端工具可能会影响性能。

将第三方客户端工具配置为使用日期、时间和时间戳的默认格式和转换设置。

如果日期、时间和时间戳格式和转换设置为用户指定的格式而不是默认格式，则第三方客户端工具可能会影响性能。

禁用调试选项或将其设置为无调试。

如果运行查询的调试选项设置为跟踪，则第三方客户端工具可能会影响性能。这可能会降低性能，因为第三方客户端工具在处理查询时会更多日志消息写入到调试文件中。

SQL 数据服务优化器级别

数据集成服务根据您配置的优化器级别优化 SQL 数据服务。希望 SQL 数据服务使用优化器级别而非普通级别时，请配置优化器级别。默认情况下，每个 SQL 数据服务都使用普通优化器级别。

要了解优化器级别如何为 SQL 数据服务创建已优化查询，请查看 SQL 数据服务的查询计划。查看查询计划时，Developer 工具根据原始查询的优化器级别和图形表示形式显示已优化查询的图形表示形式。

您可以配置以下优化器级别：

0 (无)

数据集成服务不应用任何优化。

1 (最小)

数据集成服务应用早期计划优化方法。

2 (普通)

数据集成服务将应用早期计划、早期选择、分支修剪、推入、全局谓词优化和谓词优化等方法。

3 (完全)

数据集成服务将应用基于成本的优化、早期计划、早期选择、分支修剪、谓词、推入、半联接和船舶数据中心联接优化等方法。

默认值为“2（普通）”。

您可以使用以下一种或多种方法为 SQL 数据服务配置优化器级别：

- 为 SQL 数据服务的数据预览配置优化器级别。
- 为已部署的 SQL 数据服务配置优化器级别。
- 在您针对已部署的 SQL 数据服务运行的查询的连接字符串中配置优化器级别。

为数据预览配置 SQL 数据服务优化器级别

预览 SQL 数据服务的输出时，请配置数据集成服务用于执行 SQL 查询的优化器级别。

- 1. 在 Developer 工具中，单击**运行 > 打开运行对话框**。
- 此时将显示**运行**对话框。
- 2. 单击**数据查看器配置**。
- 3. 单击**新建按钮**。
- 4. 输入数据查看器配置的名称。
- 5. 单击**高级选项卡**。
- 6. 选择优化器级别。
- 7. 单击**应用**。
- 8. 单击**关闭**

Developer 工具将创建数据查看器配置。

为已部署的 SQL 数据服务配置优化器级别

配置数据集成服务用于针对已部署的 SQL 数据服务执行 SQL 查询的优化器级别。 您可以选择通过在 SQL 数据服务连接中配置优化器级别来替代单个查询的优化器级别。

- 1. 在 Administrator 工具中，选择一项数据集成服务。
- 2. 单击**应用程序视图**。
- 3. 展开包含您要为其配置优化器级别的 SQL 数据服务的应用程序。
- 4. 选择 SQL 数据服务并编辑以下属性：

属性	说明
优化级别	数据集成服务应用到对象的优化器级别。 请输入与您要配置的优化器级别关联的数值。 您可以输入以下数值之一： <ul style="list-style-type: none">- 0. 数据集成服务不应用优化。- 1. 数据集成服务应用早期计划优化方法。- 2. 数据集成服务应用早期计划、早期选择、推入和谓词优化方法。- 3. 数据集成服务应用基于成本、早期计划、早期选择、推入、谓词和半联接优化方法。

- 5. 要替代数据集成服务用于执行查询的优化器级别，请将以下条目附加到 JDBC URL 或 ODBC 连接字符串：
SQLDataServiceOptions.optimizeLevel= <numeric_optimizer_level>.

SQL 数据服务查询计划

查看 SQL 数据服务的查询计划时，请查看原始查询的图形表示形式和已优化查询的图形表示形式。 图形表示形式介绍了数据集成服务处理查询的方式。 它包括转换以及数据集成服务处理每个转换的顺序。

Developer 工具使用您在 Developer 工具中设置的优化器级别来生成已优化查询。 已优化查询将查询显示为数据集成服务正在运行该查询。

例如，您希望查询 SQL 数据服务中的 CUSTOMERS 虚拟表。 在**数据查看器**视图中，您选择默认数据查看器配置设置，即将查询的优化器级别设置为普通。

您在**数据查看器**视图中输入以下查询：

```
select * from CUSTOMERS where CUSTOMER_ID > 150000 order by LAST_NAME
```

查看 SQL 查询计划时，Developer 工具将显示查询的以下图形表示形式：



未优化视图将显示您输入的查询。Developer 工具将 WHERE 子句显示为筛选器转换，将 ORDER BY 子句显示为排序器转换。Developer 工具使用传递表达式转换来重命名端口。

查看已优化查询时，Developer 工具将显示查询的以下图形表示形式：



已优化视图显示数据集成服务运行的查询。由于优化器级别为普通，因此，数据集成服务会将筛选条件推送到源数据对象。推送筛选条件可提高查询性能，因为它减少了数据集成服务从源数据对象中读取的行数。与未优化查询相似，Developer 工具将 ORDER BY 子句显示为排序器转换。开发人员工具使用传递表达式转换来强制使用您在逻辑转换中指定的数据类型。

查看 SQL 查询计划

显示 SQL 查询可查看您在预览虚拟表数据时输入的 SQL 查询的映射类表示形式。

1. 打开至少包含一个虚拟表的 SQL 数据服务。
2. 单击**数据查看器**视图。
3. 在**输入**窗口中输入 SQL 查询。
4. 或者，选择一个包含要应用到查询的优化器级别的数据查看器配置。
5. 单击**显示查询计划**。

Developer 工具将查询的 SQL 查询计划显示为您在**未优化**选项卡中输入时的状态。

6. 要查看已优化查询，请单击**已优化**选项卡。

Developer 工具将显示已优化 SQL 查询计划。

SQL 数据服务的内存和并发请求属性

要优化 SQL 数据服务性能，请在 Administrator 工具中配置数据集成服务的并发属性和内存属性。

下表介绍了 SQL 服务模块的“每个请求的内存上限”属性：

属性	说明
每个请求的内存上限	<p>“每个请求的内存上限”属性的行为取决于以下数据集成服务配置：</p> <ul style="list-style-type: none">- 服务在单独的本地或远程进程中运行作业，或者服务属性“内存大小上限”的值为 0（默认值）。- 在这种情况下，每个请求的内存上限是数据集成服务可以分配给一个请求中所有使用自动缓存模式的转换的最大内存量（字节数）。服务单独为具有特定缓存大小的转换分配缓存。请求所用的总内存可以超过“每个请求的内存上限”值。- 服务在数据集成服务进程中运行作业，并且服务属性“内存大小上限”的值大于 0。- 在这种情况下，每个请求的内存上限是数据集成服务可以分配给单个请求的最大内存量（字节数）。请求所用的总内存不可超过“每个请求的内存上限”值。 <p>默认值为 50,000,000。</p>

下表介绍了数据集成服务进程的“堆大小上限”属性：

属性	说明
堆大小上限	<p>分配给运行数据集成服务的 Java 虚拟机 (JVM) 的 RAM 量。使用此属性可提高性能。将以下其中一个字母附加到值可指定单位：</p> <ul style="list-style-type: none">- b 代表字节。- k 代表千字节。- m 代表兆字节。- g 代表千兆字节。 <p>默认值为 1024 MB。</p> <p>注意：当数据集成服务需要处理大量数据时，考虑增大堆大小。</p>

下表介绍了数据集成服务进程的 SQL 属性：

属性	说明
并发连接数上限	限制数据集成服务可为 SQL 数据服务建立的数据库连接数量。默认值为 100。

下表介绍了数据集成服务的执行选项：

属性	说明
按需执行池大小上限	可并发运行的最大按需作业数。作业包括数据预览、剖析作业、REST 和 SQL 查询、Web 服务请求以及从 Developer tool 运行的映射。数据集成服务接收的所有作业都会对按需池大小有影响。当有足够的资源可用时，数据集成服务会立即运行按需作业。否则，数据集成服务将拒绝作业。默认值为 10。
本地批处理执行池大小上限	可在本地环境中并发运行的最大已部署作业数。当有足够的资源可用时，数据集成服务会将本地映射作业从队列移至本地作业池。默认值为 10。

属性	说明
Hadoop 批处理执行池大小上限	可在 Hadoop 环境中并发运行的最大已部署作业数。当有足够的资源可用时，数据集成服务会将 Hadoop 作业从队列移至 Hadoop 作业池。默认值为 100。
内存大小上限	<p>当服务在数据集成服务进程中运行作业时，数据集成服务可以分配用于并行运行所有请求的最大内存量（以字节为单位）。当数据集成服务在单独的本地或远程进程中运行作业时，服务会忽略此值。如果不想限制数据集成服务可以分配的内存量，请将此属性设置为 0。</p> <p>如果该值大于 0，数据集成服务将使用该属性计算并行运行所有请求所允许使用的最大总内存。数据集成服务将按如下所示计算最大总内存： 内存大小上限 + 堆大小上限 + 加载程序组件时所需的内存</p> <p>默认值为 0。 注意: 如果您运行配置文件或数据质量映射，请将此属性设置为 0。</p>

SQL 数据服务的结果集缓存

配置结果集缓存时，数据集成服务缓存与每个 SQL 数据服务查询和 Web 服务请求关联的 DTM 进程的结果。数据集成服务在您配置的到期时间段内缓存结果。客户端在缓存过期之前执行相同的查询时，数据集成服务将返回已缓存的结果。

对于结果集缓存瓶颈，请考虑应用以下解决方案：
为 SQL 数据服务配置结果集缓存。

结果集缓存使数据集成服务能够使用 SQL 数据服务查询的已缓存结果。在短时间内运行相同查询的用户可能希望使用结果集缓存来缩短相同查询的运行时间。

启用数据集成服务以使用已缓存的结果时，数据服务性能将得到提高。但是，要进一步减少相同查询的数据服务处理时间，请分配足够的空间以在内存中存储缓存。将缓存内存量配置为等于或大于缓存结果所需的缓存内存量时，可通过降低系统 I/O 开销来提高性能。数据集成服务将缓存文件写入磁盘时，数据服务处理时间由于系统 I/O 开销而增加。

SQL 数据服务结果集缓存属性

要提高性能，可以配置数据集成服务的结果集缓存属性。还可以配置结果集缓存可用于 SQL 数据服务的毫秒数。

下表介绍了数据集成服务的结果集缓存属性：

属性	说明
文件名前缀	存储在磁盘上的所有结果集缓存文件的名称前缀。默认值为 RSCACHE。
启用加密	指示结果集缓存文件是否使用 128 位 AES 加密进行加密。有效值为 true 或 false。默认值为 true。

下表介绍的属性用于配置结果集缓存对 SQL 数据服务可用的毫秒数：

属性	说明
结果集缓存到期时间段	结果集缓存可供使用的毫秒数。如果设置为 -1，缓存将永不过期。如果设置为 0，结果集缓存将禁用。对到期时间段所做的更改不应用到现有缓存。如果您希望所有缓存使用相同的到期时间段，请在更改到期时间段后清除结果集缓存。默认值为 0。

启用 SQL 数据服务的结果集缓存

要为相同的 SQL 数据服务查询使用已缓存的结果，请将数据集成服务配置为使用结果集缓存。

1. 在 Administrator 工具中，选择一项数据集成服务。
2. 单击**进程**视图以配置结果集缓存属性。
3. 单击**应用程序**视图，然后单击 SQL 数据服务以配置结果集缓存到期属性。

在临时表中保留虚拟数据

临时表是关系数据库中用来存储中间临时数据的表。复杂查询通常需要存储大量中间数据，例如联接信息。实施临时表后，商业智能工具可以从临时表中（而不必从 SQL 数据服务中）检索此数据。这样可以提升性能。

临时表还通过两种方式提高安全性。首先，只有活动会话的用户可以访问表。另外，表在会话处于活动状态期间将持续保留，连接关闭时，数据库才丢弃表。

实施临时表

您可以使用临时表提高大型复杂查询的性能。临时表之所以能提高性能，是因为向关系数据库中的临时表查询数据要比重复向 SQL 数据服务查询相同的数据集速度更快。

要实施临时表来提高性能，需要 Informatica 管理员和商业智能工具开发人员执行相应的操作。

首先，Informatica 管理员创建一个关系数据库连接，并将数据集成服务配置为使用该连接。

然后，商业智能工具（例如 IBM Cognos 或 SAP Business Objects）开发人员在商业智能工具与 Informatica SQL 数据服务之间建立连接。连接使用 Informatica ODBC 或 JDBC 驱动程序。

当这些连接处于活动状态时，商业智能工具可以创建并使用临时表处理大量中间数据。

第 9 章

Web 服务优化

本章包括以下主题：

- [Web 服务优化概览, 62](#)
- [优化 HTTP 请求, 63](#)
- [Web 服务消息压缩, 63](#)
- [Web 服务优化器级别, 63](#)
- [内存和并发请求的 Web 服务属性, 65](#)
- [用于配置活动 DTM 实例的 Web 服务属性, 67](#)
- [Web 服务结果集缓存, 67](#)
- [Web 服务日志管理, 68](#)

Web 服务优化概览

数据集成服务运行 Web 服务请求时，您可以优化 Web 服务以提高性能。要管理内存并处理并发 Web 服务请求，请调整数据集成服务。要提高 Web 服务性能，请使用 Web 服务消息压缩、优化 HTTP 请求、配置数据对象和结果集缓存以及配置错误日志级别。

使用以下优化技术可优化 Web 服务：

- 优化 HTTP 请求。
- 压缩 Web 服务消息。
- 配置 Web 服务优化器级别。
- 为数据集成进程配置并发请求和内存的 Web 服务属性。
- 将数据集成服务配置为保持 DTM 进程处于活动状态，以便其能够处理多个 Web 服务请求。
- 为 Web 服务配置数据对象缓存。
- 为 Web 服务配置结果集缓存。
- 配置 Web 服务运行时错误日志级别。

相关主题：

- [“数据对象缓存” 页面上 51](#)

优化 HTTP 请求

优化 HTTP 请求可减少 Web 服务器的请求数。

对于 HTTP 请求瓶颈，请考虑应用以下解决方案：

缩短 Web 服务客户端的 HTTP 套接字超时。

套接字超时设置客户端在 HTTP 请求超时之前等待的时间。如果套接字超时值非常大，Web 服务客户端可能会挂起。

Web 服务消息压缩

您可以通过压缩传递到提供程序以及来自提供程序的大型 Web 消息来优化 Web 服务性能。

对于 Web 服务消息瓶颈，请考虑应用以下解决方案：

为 Web 服务客户端启用 SOAP 消息压缩。

通过 SOAP 消息压缩，Web 服务能够压缩 Web 服务以接收压缩的 Web 服务客户端消息。Web 服务可以接受来自 Web 服务客户端并使用 GZip 进行压缩的 SOAP 消息。

数据集成服务接收来自 Web 服务的响应时，将检查 SOAP 消息中的内容编码 HTTP 表头并对该消息进行解码。

Web 服务优化器级别

数据集成服务根据您配置的优化器级别优化 Web 服务。希望 Web 服务使用除普通级别外的优化器级别时，请配置优化器级别。默认情况下，每个 Web 服务都使用普通优化器级别。

您可以选择以下优化器级别之一：

0 (无)

数据集成服务不应用任何优化。

1 (最小)

数据集成服务应用早期计划优化方法。

2 (普通)

数据集成服务将应用早期计划、早期选择、分支修剪、推入、全局谓词优化和谓词优化等方法。

3 (完全)

数据集成服务将应用基于成本的优化、早期计划、早期选择、分支修剪、谓词、推入、半联接和船舶数据中心联接优化等方法。

默认值为“2（普通）”。

您可以使用以下一种或多种方法为 Web 服务配置优化器级别：

- 在部署到数据集成服务之前为 Web 服务的数据预览配置优化器级别。
- 为特定数据集成服务上运行的已部署 Web 服务配置优化级别。
- 在已部署 Web 服务的 Web 服务请求的表头中配置优化器级别。

为数据预览配置 Web 服务优化器级别

配置数据集成服务用于预览 Web 服务输出的优化器级别。

1. 在 Developer 工具中，单击**运行 > 打开运行对话框**。
此时将显示**运行对话框**。
2. 单击 **Web 服务配置**。
3. 单击**新建按钮**。
4. 输入 Web 服务配置的名称。
5. 单击**高级选项卡**。
6. 选择优化器级别。
7. 单击**应用**。
8. 单击**关闭**

Developer 工具将创建 Web 服务配置。

运行数据查看器以预览操作映射的输出时，请选择包含要使用的优化器级别的 Web 服务配置。

为已部署的 Web 服务配置优化器级别

配置数据集成服务用于运行已部署 Web 服务的优化器级别。 您可以选择通过在 Web 服务 SOAP 请求的 HTTP 表头中配置优化器级别来替代单个请求的优化器级别。

1. 在 Administrator 工具中，选择一项数据集成服务。
2. 单击**应用程序视图**。
3. 展开包含您要为其配置优化器级别的 Web 服务的应用程序。
4. 选择 Web 服务并编辑以下属性：

属性	说明
优化级别	数据集成服务应用到对象的优化器级别。 请输入与您要配置的优化器级别关联的数值。 您可以输入以下数值之一： <ul style="list-style-type: none">- 0. 数据集成服务不应用优化。- 1. 数据集成服务应用早期计划优化方法。- 2. 数据集成服务应用早期计划、早期选择、推入和谓词优化方法。- 3. 数据集成服务应用基于成本、早期计划、早期选择、推入、谓词和半联接优化方法。

5. 要替代 Web 服务请求的 Web 服务优化级别，请在 Web 服务 SOAP 请求的 HTTP 表头中包括以下条目：
`WebServiceOptions.optimizeLevel= <numeric_optimizer_level>`.

内存和并发请求的 Web 服务属性

要优化 Web 服务性能，请在 Administrator 工具中配置数据集成服务和每个 Web 服务的并发和内存属性。

下表介绍了 Web 服务模块的“每个请求的内存上限”属性：

属性	说明
每个请求的内存上限	<p>“每个请求的内存上限”属性的行为取决于以下数据集成服务配置：</p> <ul style="list-style-type: none">- 服务在单独的本地或远程进程中运行作业，或者服务属性“内存大小上限”的值为 0（默认值）。在这种情况下，每个请求的内存上限是数据集成服务可以分配给一个请求中所有使用自动缓存模式的转换的最大内存量（字节数）。服务单独为具有特定缓存大小的转换分配缓存。请求所用的总内存可以超过“每个请求的内存上限”值。- 服务在数据集成服务进程中运行作业，并且服务属性“内存大小上限”的值大于 0。在这种情况下，每个请求的内存上限是数据集成服务可以分配给单个请求的最大内存量（字节数）。请求所用的总内存不可超过“每个请求的内存上限”值。 <p>默认值为 50,000,000。</p>

下表介绍了数据集成服务的执行选项：

属性	说明
内存大小上限	<p>当服务在数据集成服务进程中运行作业时，数据集成服务可以分配用于并行运行所有请求的最大内存量（以字节为单位）。当数据集成服务在单独的本地或远程进程中运行作业时，服务会忽略此值。如果不想限制数据集成服务可以分配的内存量，请将此属性设置为 0。</p> <p>如果该值大于 0，数据集成服务将使用该属性计算并行运行所有请求所允许使用的最大总内存。数据集成服务将按如下所示计算最大总内存：</p> <p>内存大小上限 + 堆大小上限 + 加载程序组件时所需的内存</p> <p>默认值为 0。</p> <p>注意: 如果您运行配置文件或数据质量映射，请将此属性设置为 0。</p>

下表介绍了数据集成服务进程的 HTTP 配置属性：

属性	说明
积压请求数上限	可排队等待此数据集成服务进程的 HTTP 或 HTTPS 连接数上限。默认值为 100。
并发请求数上限	可用于此数据集成服务进程的 HTTP 或 HTTPS 连接数上限。最小值为 4。默认值为 200。 注意: 对于 Web 服务，此属性将影响数据集成服务在将请求发送到数据集成服务积压之前接受的 Web 服务请求数。

下表介绍了您可以为数据集成服务进程配置的“堆大小上限”属性：

属性	说明
堆大小上限	分配给运行数据集成服务的 Java 虚拟机 (JVM) 的 RAM 量。使用此属性可提高性能。将以下其中一个字母附加到值可指定单位： <ul style="list-style-type: none">- b 代表字节。- k 代表千字节。- m 代表兆字节。- g 代表千兆字节。 默认值为 1024 MB。 注意: 当数据集成服务需要处理大量数据时，考虑增大堆大小。

并发 Web 服务请求的数据集成服务配置示例

配置数据集成服务处理并发 Web 服务请求的方式时，请验证并发请求数上限的值是否与 Web 服务和数据集成服务进程的值相同。

例如，在以下配置中，数据集成服务接受 200 个并发 HTTP 请求，但仅接受 10 个 Web 服务并发请求：

属性类型	属性名称	配置
数据集成服务进程	并发请求数上限	200
数据集成服务进程	积压请求数上限	500
数据集成服务	按需执行池大小上限	100
Web 服务	并发请求数上限	10

数据集成服务接收 20 个 Web 服务请求时，10 个 Web 服务请求将失败，因为 Web 服务只能接收 10 个并发请求。

要避免在 Web 服务达到并发请求数上限时 Web 服务请求失败，请为数据集成服务进程和 Web 服务配置相同的上限值。发送到数据集成服务的请求数超过最大并发请求值时，其他请求将保留在积压中，直至数据集成服务进程能够用于处理请求。

用于配置活动 DTM 实例的 Web 服务属性

要提高性能，您可以将数据集成服务配置为保持 DTM 实例处于活动状态，以便其能够处理多个 Web 服务请求。您可以在 Administrator 工具中为数据集成服务配置“DTM 保持活动的时间”属性。

下表介绍了“DTM 保持活动的时间”属性：

属性	说明
DTM 保持活动的时间	<p>DTM 实例在完成最后一个请求之后保持打开状态的毫秒数。针对相同的操作发出的 Web 服务请求可以重用处于打开状态的实例。当处理请求所需的时间少于 DTM 实例的初始化时间时，使用“保持活动的时间”属性来提高性能。如果请求失败，DTM 实例将终止。</p> <p>默认值为 5000。</p> <p>注意: 使用现有 DTM 实例提高性能的功能。DIS 需要额外的资源才能为每个请求启动 DTM 实例。保持 DTM 处于活动状态会消耗内存。因此，用户在配置此选项时应考虑内存消耗。</p>

Web 服务结果集缓存

配置结果集缓存时，数据集成服务将缓存与每个 Web 服务请求关联的 DTM 进程的结果。数据集成服务在您配置的到期时间段内缓存结果。外部客户端在缓存过期之前运行相同的请求时，数据集成服务将返回已缓存的结果。

对于结果集缓存瓶颈，请考虑应用以下解决方案：

为 Web 服务配置结果集缓存。

结果集缓存使数据集成服务能够使用 Web 服务请求的已缓存结果。在短时间内运行相同查询的用户可能希望使用结果集缓存来缩短相同查询的运行时间。

Web 服务使用“Web 服务安全性”时，数据集成服务按用户存储 Web 服务的结果集缓存。数据集成服务按在 Web 服务请求的用户名令牌中提供的用户名存储缓存。数据集成服务按用户缓存结果时，仅将已缓存的结果返回给发送 Web 服务请求的用户。

为 Web 服务启用结果集缓存

要为相同的 Web 服务请求使用已缓存的结果，请将数据集成服务配置为使用结果集缓存。

1. 在 Administrator 工具中，选择一项数据集成服务。
2. 单击**进程**视图以配置结果集缓存属性。
3. 依次单击**应用程序**视图、Web 服务以及在 Web 服务操作属性中配置缓存到期时间段的操作。如果您希望数据集成服务按用户缓存结果，请在 Web 服务属性中启用“WS-Security”。
4. 要在将 Web 服务操作配置为缓存结果集时为 Web 服务请求禁用结果集缓存，请在 SOAP 请求的 HTTP 表头中包括以下语法：

```
WebServiceOptions.disableResultSetCache=true
```

Web 服务日志管理

数据集成服务写入并维护大量日志文件时，系统 I/O 性能可能会降低。数据集成服务根据您配置的跟踪级别生成 Web 服务运行时日志。请考虑管理数据集成服务写入和维护的日志文件数。

对于 Web 服务日志瓶颈，请考虑应用以下解决方案：

将 Web 服务跟踪级别设置为“关闭”。

为已部署的 Web 服务配置 Web 服务属性时，您可以指定日志跟踪级别。跟踪级别决定数据集成服务写入运行时日志位置的日志类型。默认 Web 服务跟踪级别为“信息”。跟踪级别设置为“最细致”或“全部”时，性能可能会降低，因为数据集成服务会将更多日志写入日志文件。Web 服务使用“HTTPS”或“WS-Security”时，将跟踪级别设置为“最细致”或“全部”对性能的影响最大。

不再需要的存档日志文件。

如果存储的日志文件过多，则会影响系统 I/O。默认情况下，数据集成服务在以下目录中写入 Web 服务运行时日志： <InformaticaInstallationDir>/tomcat/bin/disLogs/ws

注意：如果您删除了 ws 文件夹以清空日志，则必须重新创建 ws 文件夹。删除并重新创建 ws 文件夹之前，请停止数据集成服务。

在数据集成服务中启用“跳过日志”属性

启用**跳过日志**属性可防止数据集成服务在 Web 服务请求成功完成时生成日志文件。必须在数据集成服务中将 Web 服务的跟踪级别设置为 INFO 或更高。

第 10 章

连接优化

本章包括以下主题：

- [连接优化概览, 69](#)
- [连接池, 69](#)
- [数据库网络包大小, 70](#)

连接优化概览

您可以优化连接以提高性能。您可以管理数据库连接的空闲连接实例的池。您可以增大网络包大小以允许更大型的数据包同时跨网络。

使用以下技术可优化连接：

- 优化连接池。
- 优化数据库网络包大小。

连接池

连接池是指用于缓存数据集成服务使用的数据库连接信息的框架。连接池通过重用已缓存的连接信息来提高性能。

对于连接瓶颈，请考虑应用以下解决方案：

为数据库连接启用连接池。

启用连接池可优化连接性能。您可以管理数据库连接的空闲连接实例。连接池根据您配置的池属性保留空闲连接实例。您可以调整空闲连接数上限和下限以及空闲连接的等待时间上限。

连接对象中的池属性

您可以编辑数据库连接的池视图中的连接池属性。

连接池库的数量取决于正在运行的数据集成服务进程或 DTM 进程的数量。每个数据集成服务进程或 DTM 进程都会维护各自的连接池库。池属性的值适用于每个连接池库。

例如，如果您将连接数上限设置为 15，则每个连接池库在池中最多可以有 15 个空闲连接。如果数据集成服务在单独的本地进程中运行作业，并且有三个 DTM 进程正在运行，则最多可以有 45 个空闲连接实例。

要减少空闲连接实例总数，请将连接数下限设置为 0，并减少每个数据库连接的空闲时间上限。

下表介绍了您可以在数据库连接的池视图中编辑的数据库连接池属性：

启用连接池

启用连接池。启用连接池时，每个连接池都在内存中保留空闲连接实例。要删除空闲连接的池，必须重新启动数据集成服务。

如果连接池已禁用，DTM 进程或数据集成服务进程将停止所有池活动。DTM 进程或数据集成服务进程在每次处理作业时创建一个连接实例。完成处理该作业后，将删除对应的实例。

默认为 DB2 for i5/OS、DB2 for z/OS、IBM DB2、Microsoft SQL Server、Oracle 和 ODBC 连接启用连接池。默认为 Adabas、IMS、Sequential 和 VSAM 连接禁用连接池。

连接数下限

池在达到空闲时间上限后为数据库连接维护的空闲连接实例数下限。请将此值设置为等于或小于空闲连接实例数上限。默认值为 0。

连接数上限

池在达到空闲时间上限之前为数据库连接维持的空闲连接实例数上限。请将此值设置为大于空闲连接实例数下限。默认值为 15。

空闲时间上限

超出连接实例数下限的连接实例可以保持空闲状态的秒数，超过此时间后连接池会将其删除。连接实例不超出空闲连接实例数下限时，连接池将忽略此空闲时间。默认值为 120。

数据库网络包大小

如果您从 Oracle、Sybase ASE 或 Microsoft SQL Server 目标中读取数据或将数据写入这些目标，则可以根据您读取或写入的数据库增大网络包大小来提高性能。增大网络包大小可允许更大型的数据包同时跨网络。

对于数据库网络包大小瓶颈，请考虑应用以下解决方案：

增大 Oracle 数据库的数据库网络包大小。

您可以在 listener.ora 和 tnsnames.ora 中增大数据库服务器网络包大小。有关增大包大小的其他信息，请参阅数据库文档（如有需要）。

增大 Sybase ASE 数据库的数据库网络包大小。

有关如何增大包大小的信息，请参阅数据库文档。您还必须在数据集成服务的关系连接对象中更改 Sybase ASE 的包大小，以反映数据库服务器包大小。

增大 Microsoft SQL Server 数据库的数据库网络包大小。

有关如何增大包大小的信息，请参阅数据库文档。您还必须在数据集成服务的关系连接对象中更改 Microsoft SQL Server 的包大小，以反映数据库服务器包大小。

索引

B

- 半连接优化
 - 说明 [38](#)
- 表达式优化
 - 映射优化 [22](#)
- 并发请求
 - SQL 数据服务 [59](#)
 - Web 服务 [65](#)

C

- 查询视图
 - 配置提示 [18](#)
- 查询优化
 - 源优化 [16](#)
- 查找转换
 - 转换优化 [26](#)
- 船舶数据中心连接优化
 - 说明 [37](#)
- 错误跟踪
 - 映射优化 [43](#)
- 错误跟踪级别
 - Web 服务日志管理 [68](#)

D

- 第三方客户端工具
 - 运行时优化 [55](#)

F

- 分区
 - 多个 CPU [44](#)
 - 优化 [44](#)
 - 优化目标数据库 [47](#)
 - 优化平面文件目标 [45](#)
 - 优化平面文件源 [45](#)
 - 优化源数据库 [46](#)
 - 优化转换 [47](#)
- 分支修剪优化
 - 说明 [39](#)
- 副作用
 - 说明 [31](#)
 - SQL 转换 [29](#)
 - Web 服务使用者转换 [32](#)

H

- 汇总器转换
 - 转换优化 [21](#)

- 活动 DTM 实例
 - Web 服务 [67](#)

J

- Java 转换
 - 转换优化 [24](#)
- JDBC 驱动程序
 - 运行时优化 [56](#)
- 监视统计信息
 - 运行时优化 [49](#)
- 结果集缓存
 - 结果集缓存属性 [60](#)
 - 为 SQL 数据服务启用结果集缓存 [61](#)
- 结果集缓存属性
 - 运行时优化 [60](#)
- 基于成本的优化
 - 说明 [37](#)

L

- 连接池
 - 连接优化 [69](#)
 - 属性 [69](#)
- 联接器转换
 - 转换优化 [26](#)
- 连接优化
 - 连接池 [69](#)
 - 数据库网络包大小 [70](#)
- 临时表
 - 说明 [61](#)
- 逻辑数据对象
 - 在数据库中缓存 [51](#)

M

- 目标优化
 - Oracle 数据库优化 [14](#)
 - 批量加载 [14](#)
 - 平面文件目标 [13](#)
 - 数据库检查点时间间隔 [14](#)

N

- 内存分配
 - 活动 DTM 实例 [67](#)
 - SQL 数据服务 [59](#)
 - Web 服务 [67](#)
 - 并发请求 [59](#)

O

Oracle 数据库优化
 目标优化 [14](#)
 源优化 [20](#)

P

排序器转换
 转换优化 [28](#)
批量加载
 目标优化 [14](#)
瓶颈
 在 UNIX 中 [11](#)
 在 Windows 中 [11](#)
平面文件
 优化目标以进行分区 [45](#)
 优化源以进行分区 [45](#)
平面文件目标
 目标优化 [13](#)
平面文件源
 源优化 [15](#)
普通优化级别
 说明 [35](#)

S

筛选器端口
 Web 服务使用者转换 [32](#)
筛选器优化
 映射优化 [42](#)
数据对象缓存
 表数据类型 [51](#)
 配置 [51](#)
 说明 [51](#)
 索引缓存 [51](#)
 用户管理的表 [51](#)
 优化 [53](#)
数据集成服务
 SQL 数据服务结果集缓存 [60](#)
 Web 服务结果集缓存 [67](#)
数据库
 优化目标以进行分区 [47](#)
 优化源以进行分区 [46](#)
数据库检查点时间间隔
 目标优化 [14](#)
数据库提示
 在 Developer 工具中输入 [18](#)
数据库网络包大小
 连接优化 [70](#)
数据类型转换优化
 映射优化 [42](#)
SQL 查询计划
 查看 [58](#)
SQL 数据服务结果集缓存
 数据集成服务 [60](#)
SQL 数据服务优化
 第三方客户端工具 [55](#)
 JDBC 驱动程序 [56](#)
SQL 提示
 在 Developer 工具中输入 [18](#)
SQL 转换
 推入优化 [29](#)
 推入优化属性 [30](#)
 早期选择优化 [29](#)
 转换优化 [29](#)

SQL 数据服务
 内存分配 [59](#)

T

条件筛选器
 源优化 [16](#)
提示
 查询视图 [18](#)
推入优化
 说明 [40](#)
 SQL 转换 [29](#)
 Web 服务使用者转换 [32](#)
 在 SQL 转换中启用 [30](#)

U

UNIX
 系统瓶颈 [11](#)

W

Web 服务结果集缓存
 数据集成服务 [67](#)
Web 服务日志管理
 错误跟踪级别 [68](#)
Web 服务使用者转换
 启用推入优化 [33](#)
 筛选器优化 [32](#)
 推入优化 [32](#)
 早期选择优化 [32](#)
 转换优化 [31](#)
Web 服务消息压缩
 Web 服务优化 [63](#)
Web 服务优化
 Web 服务消息压缩 [63](#)
 优化 HTTP 请求 [63](#)
Web 服务
 内存分配 [67](#)
 并发请求 [65](#)
为 SQL 数据服务启用结果集缓存
 结果集缓存 [61](#)
Windows
 瓶颈 [11](#)
完整优化级别
 说明 [35](#)

X

下推优化
 说明 [40](#)
性能优化
 半联接优化方法 [38](#)
 船舶数据中心联接优化方法 [37](#)
 分支修剪优化方法 [39](#)
 进程概览 [9](#)
 基于成本的优化方法 [37](#)
 全局谓词优化方法 [39](#)
 推入优化方法 [40](#)
 谓词优化方法 [36](#)
 下推优化方法 [40](#)
 优化方法 [35](#)
 早期计划优化方法 [36](#)
 优化级别 [35](#)

- 性能优化 (续)
 - 早期选择优化方法 [39](#)
- 系统
 - UNIX 中的瓶颈, 标识 [11](#)
 - Windows 中的瓶颈, 标识 [11](#)
- 系统优化
 - 运行时优化 [53](#)
- 选择相异
 - 源优化 [17](#)
- 虚拟表
 - 在数据库中缓存 [51](#)
- 下推优化方法
 - 源下推 [41](#)
 - 完整下推 [40](#)

Y

- 一次性读取
 - 映射优化 [42](#)
- 映射
 - 全局谓词优化方法 [39](#)
 - 谓词优化方法 [36](#)
 - 已分区优化 [44](#)
 - 优化方法 [35](#)
- 映射优化
 - 表达式优化 [22](#)
 - 错误跟踪 [43](#)
 - 筛选器优化 [42](#)
 - 数据类型转换优化 [42](#)
 - 一次性读取 [42](#)
- 有副作用
 - 转换属性说明 [31](#)
- 优化
 - 半联接优化方法 [38](#)
 - 船舶数据中心联接优化方法 [37](#)
 - 分支修剪优化方法 [39](#)
 - 副作用 [31](#)
 - 基于成本的优化方法 [37](#)
 - 推入优化方法 [40](#)
 - 下推优化方法 [40](#)
 - 映射性能方法 [35](#)
 - 早期计划优化方法 [36](#)
 - 早期选择优化方法 [39](#)
- 优化 HTTP 请求
 - Web 服务优化 [63](#)
- 源优化
 - 查询优化 [16](#)

- 源优化 (续)
 - Oracle 数据库优化 [20](#)
 - 平面文件源 [15](#)
 - 条件筛选器 [16](#)
 - 选择相异 [17](#)
 - 约束 [18](#)
 - 自定义数据对象 [19](#)
- 约束
 - 配置约束 [19](#)
 - 源优化 [18](#)
- 运行时优化
 - 监视统计信息 [49](#)
 - 系统优化 [53](#)
- 优化级别
 - 说明 [35](#)

Z

- 早期计划优化
 - 说明 [36](#)
- 早期选择优化
 - SQL 转换 [29](#)
 - Web 服务使用者转换 [32](#)
 - 说明 [39](#)
- 转换
 - 针对分区进行优化 [47](#)
- 转换错误消除
 - 转换优化 [31](#)
- 转换缓存
 - 转换优化 [30](#)
- 转换优化
 - 查找转换 [26](#)
 - 汇总器转换 [21](#)
 - Java 转换 [24](#)
 - 联接器转换 [26](#)
 - 排序器转换 [28](#)
 - SQL 转换 [29](#)
 - Web 服务使用者转换 [31](#)
 - 转换错误消除 [31](#)
 - 转换缓存 [30](#)
- 自定义数据对象
 - 源优化 [19](#)
- 最大并行数
 - 增加 [45](#)
- 最小优化级别
 - 说明 [35](#)