



Informatica® PowerCenter
10.4.0

性能优化指南

Informatica PowerCenter 性能优化指南

10.4.0

2019 年 12 月

© 版权所有 Informatica LLC 2000, 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、Informatica 标志和 PowerCenter 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/或文档的某些部分受第三方版权制约，包括但不限于：版权所有 DataDirect Technologies。保留所有权利。版权所有 (C) Sun Microsystems。保留所有权利。版权所有 (C) RSA Security Inc. 保留所有权利。版权所有 (C) Ordinal Technology Corp. 保留所有权利。版权所有 (C) Aandacht c.v. 保留所有权利。版权所有 Genivia, Inc. 保留所有权利。版权所有 Isomorphic Software。保留所有权利。版权所有 (C) Meta Integration Technology, Inc. 保留所有权利。版权所有 (C) Intalio。保留所有权利。版权所有 (C) Oracle。保留所有权利。版权所有 (C) Adobe Systems Incorporated。保留所有权利。版权所有 (C) DataArt, Inc. 保留所有权利。版权所有 (C) ComponentSource。保留所有权利。版权所有 (C) Microsoft Corporation。保留所有权利。版权所有 (C) Rogue Wave Software, Inc. 保留所有权利。版权所有 (C) Teradata Corporation。保留所有权利。版权所有 (C) Yahoo! Inc. 保留所有权利。版权所有 (C) Glyph & Cog, LLC。保留所有权利。版权所有 (C) Thinkmap, Inc. 保留所有权利。版权所有 (C) Clearpace Software Limited。保留所有权利。版权所有 (C) Information Builders, Inc. 保留所有权利。版权所有 (C) OSS Nokalva, Inc. 保留所有权利。版权所有 Edifecs, Inc. 保留所有权利。版权所有 Cleo Communications, Inc. 保留所有权利。版权所有 (C) International Organization for Standardization 1986。保留所有权利。版权所有 (C) ej-technologies GmbH。保留所有权利。版权所有 (C) Jaspersoft Corporation。保留所有权利。版权所有 (C) International Business Machines Corporation。保留所有权利。版权所有 (C) yWorks GmbH。保留所有权利。版权所有 (C) Lucent Technologies。保留所有权利。版权所有 (C) University of Toronto。保留所有权利。版权所有 (C) Daniel Veillard。保留所有权利。版权所有 (C) Flexera Software。保留所有权利。版权所有 (C) Jinfonet Software。保留所有权利。版权所有 (C) Apple Inc. 保留所有权利。版权所有 (C) Telerik Inc. 保留所有权利。版权所有 (C) BEA Systems。保留所有权利。版权所有 (C) PDFlib GmbH。保留所有权利。版权所有 (C) Orientation in Objects GmbH。保留所有权利。版权所有 (C) Tanuki Software, Ltd. 保留所有权利。版权所有 (C) Ricebridge。保留所有权利。版权所有 (C) Sencha, Inc. 保留所有权利。版权所有 (C) Scalable Systems, Inc. 保留所有权利。版权所有 (C) jQWidgets。保留所有权利。版权所有 (C) Tableau Software, Inc. 保留所有权利。版权所有 (C) MaxMind, Inc. 保留所有权利。版权所有 (C) TMat Software s.r.o. 保留所有权利。版权所有 (C) MapR Technologies Inc. 保留所有权利。版权所有 (C) Amazon Corporate LLC。保留所有权利。版权所有 (C) Highsoft。保留所有权利。版权所有 (C) Python Software Foundation。保留所有权利。版权所有 (C) BeOpen.com。保留所有权利。版权所有 (C) CNRI。保留所有权利。

本产品包括由 Apache Software Foundation (<http://www.apache.org/>) 开发的软件和/或在不同 Apache 许可证版本（以下简称“许可证”）下许可的其他软件。您可从 <http://www.apache.org/licenses/> 获取这些许可证的副本。除非适用法律要求或者有相应书面协议，否则依据这些“许可证”分发的软件以“原样”提供，不附带任何明示或暗示的担保或条件。请参阅“许可证”中规定的具体语言管理权限和限制。

本产品包括由 Mozilla (<http://www.mozilla.org/>) 开发的软件、由 JBoss Group, LLC 开发的软件（版权所有 JBoss Group, LLC 保留所有权利）、由 Bruno Lowagie 和 Paulo Soares 开发的软件（版权所有 (C) 1999-2006 Bruno Lowagie 和 Paulo Soares）以及在 <http://www.gnu.org/licenses/lgpl.html> 网站上的不同版本 GNU Lesser General 公共许可协议下许可的软件。这些材料由 Informatica 按“原样”免费提供，不附带任何明示或暗示的担保，包括但不限于适销性和特定用途适用性的暗示担保。

本产品包括 ACE(TM) 和 TAO(TM) 软件，这些软件版权归 Douglas C. Schmidt 及其在华盛顿大学、加利福尼亚大学欧芬分校以及范德堡大学的研发团队所有（版权所有 (C) 1993-2006，保留所有权利）。

本产品包括由 OpenSSL Project 开发并在 OpenSSL Toolkit（版权所有 OpenSSL Project。保留所有权利）中使用的软件，该软件的再分发受 <http://www.openssl.org> 和 <http://www.openssl.org/source/license.html> 上规定条款之制约。

本产品包括 Curl 软件，版权所有 1996-2013，Daniel Stenberg <daniel@haxx.se>。保留所有权利。有关该软件的权限和限制受 <http://curl.haxx.se/docs/copyright.html> 上规定条款之制约。允许出于任何目的以免费或收费形式使用、复制、修改和分发该软件，但前提是所有副本均应注明上述版权声明以及本许可声明。

本产品包括由 MetaStuff, Ltd. 开发的软件，版权所有 2001-2005 ((C)) MetaStuff, Ltd. 保留所有权利。有关该软件的权限和限制受 <http://www.dom4j.org/license.html> 上规定条款之制约。

本产品包括由 Per Bothner 开发的软件，版权所有 (C) 1996-2006 Per Bothner。保留所有权利。<http://www.gnu.org/software/kawa/Software-License.html> 上的许可证中规定了您使用这些材料的权利。

本产品包括 OSSP UUID 软件，版权所有 (C) 2002 Ralf S. Engelschall，版权所有 (C) 2002 OSSP Project，版权所有 (C) 2002 Cable & Wireless Deutschland。有关该软件的权限和限制受 <http://www.opensource.org/licenses/mit-license.php> 上规定条款之制约。

本产品包括由 Boost (<http://www.boost.org/>) 开发的软件或在 Boost 软件许可证下许可的软件。有关该软件的权限和限制受 http://www.boost.org/LICENSE_1_0.txt 上规定条款之制约。

本产品包括由 University of Cambridge 开发的软件，版权所有 (C) 1997-2007 University of Cambridge。有关该软件的权限和限制受 <http://www.pcre.org/license.txt> 上规定条款之制约。

本产品包括由 The Eclipse Foundation 开发的软件，版权所有 (C) 2007 The Eclipse Foundation。保留所有权利。有关该软件的权限和限制受 <http://www.eclipse.org/org/documents/epl-v10.php> 和 <http://www.eclipse.org/org/documents/edl-v10.php> 上规定条款之制约。

本产品包括在 <http://www.tcl.tk/software/tcltk/license.html>、<http://www.bosrup.com/web/overlib/?License>、<http://www.stlport.org/doc/license.html>、<http://asm.ow2.org/license.html>、<http://www.cryptix.org/LICENSE.TXT>、<http://hsqldb.org/web/hsqllicense.html>、<http://httpunit.sourceforge.net/doc/license.html>、<http://jung.sourceforge.net/license.txt>、http://www.gzip.org/zlib/zlib_license.html、<http://www.openldap.org/software/release/license.html>、<http://www.libssh2.org>、<http://slf4j.org/license.html>、<http://www.sente.ch/software/OpenSourceLicense.html>、<http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>、<http://antlr.org/license.html>、<http://aopalliance.sourceforge.net/>、<http://www.bouncycastle.org/licence.html>、<http://www.jgraph.com/jgraphdownload.html>、<http://www.jcraft.com/jsch/LICENSE.txt>、http://jotm.objectweb.org/bsd_license.html、<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>、<http://www.slf4j.org/license.html>、<http://nanoxml.sourceforge.net/orig/copyright.html>、<http://www.json.org/license.html>、<http://forge.ow2.org/projects/javaservice/>、<http://www.postgresql.org/about/license.html>、<http://www.sqlite.org/copyright.html>、<http://www.tcl.tk/software/tcltk/license.html>、<http://www.jaxen.org/faq.html>、<http://www.jdom.org/docs/faq.html>、<http://www.slf4j.org/license.html>、<http://www.iodbc.org/dataspace/iodbc/wiki/IODBC/License>、<http://www.keplerproject.org/md5/license.html>、<http://www.toedter.com/en/jcalendar/license.html>、<http://www.edankert.com/bounce/index.html>、<http://www.net-snmp.org/about/license.html>、<http://www.openmdx.org/#FAQ>、http://www.php.net/license/3_01.txt、<http://srp.stanford.edu/license.txt>、<http://www.schneier.com/blowfish.html>、<http://www.jmock.org/license.html>、<http://xsom.java.net>、<http://benalman.com/about/license/>、<https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>、<http://www.h2database.com/html/license.html#summary>、<http://jsoncpp.sourceforge.net/LICENSE>、<http://jdbc.postgresql.org/license.html>、<http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>、<https://github.com/rantav/hector/blob/master/LICENSE>、<http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>、<http://jibx.sourceforge.net/jibx-license.html>、<https://github.com/Lyokato/libgeohash/blob/master/LICENSE>、<https://github.com/hjiang/jsonxx/blob/master/LICENSE>、<https://code.google.com/p/lz4/>、

<https://github.com/jedisct1/libsodium/blob/master/LICENSE>、<http://one-jar.sourceforge.net/index.php?page=documents&file=license>、<https://github.com/EsotericSoftware/kryo/blob/master/license.txt>、<http://www.scala-lang.org/license.html>、<https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>、<http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>、<https://aws.amazon.com/asl/>、<https://github.com/twbs/bootstrap/blob/master/LICENSE> 和 <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt> 下许可的软件。

本产品包括在 Academic 免费许可证 (<http://www.opensource.org/licenses/afl-3.0.php>)、通用开发和分发许可证 (<http://www.opensource.org/licenses/cddl1.php>)、通用公共许可证 (<http://www.opensource.org/licenses/cpl1.0.php>)、Sun Binary Code 许可协议补充许可条款、BSD 许可证 (<http://www.opensource.org/licenses/bsd-license.php>)、新 BSD 许可证 (<http://opensource.org/licenses/BSD-3-Clause>)、MIT 许可证 (<http://www.opensource.org/licenses/mit-license.php>)、Artistic 许可证 (<http://www.opensource.org/licenses/artistic-license-1.0>) 以及原始开发者公共许可证版本 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>) 下许可的软件。

本产品包括由 Joe Walnes 和 XStream Committers 开发的软件，版权所有 (C) 2003-2006 Joe Walnes, 2006-2007 XStream Committers。保留所有权利。有关该软件的权限和限制受 <http://xstream.codehaus.org/license.html> 上规定条款之制约。本产品包括由 Indiana University Extreme! Lab 开发的软件。有关详细信息，请访问 <http://www.extreme.indiana.edu/>。

本产品包括软件版权所有 (c) 2013 Frank Balluffi 和 Markus Moeller。保留所有权利。有关此软件的权限和限制受 MIT 许可证上规定条款之制约。

请参阅位于以下位置的专利：<https://www.informatica.com/legal/patents.html>。

免责声明：Informatica LLC 以“原样”提供本文档，不附带任何明示或暗示的担保，包括但不限于非侵权、适销性或特定用途适用性的暗示担保。Informatica LLC 不保证本软件和文档中没有错误。本软件或文档中提供的信息可能包括技术上的不准确性或排字错误。本软件和文档中包含的信息随时可能更改，恕不另行通知。

声明

本 Informatica 产品（以下称“软件”）包括由 Progress Software Corporation 的运营公司 DataDirect Technologies（以下称“DataDirect”）提供的某些驱动程序（以下称“DataDirect 驱动程序”），受以下条款和条件制约：

1. DataDirect 驱动程序以“原样”提供，不附带任何明示或暗示的担保，包括但不限于适销性、特定用途适用性以及非侵权的暗示担保。
2. 在任何情况下，DataDirect 或其第三方供应商均不对最终用户客户承担因使用 ODBC 驱动程序而引起的任何直接、间接、偶发、特殊、继发或其他损害赔偿的责任，无论是否已提前告知该种损害的可能性。这些限制适用于所有诉因，包括但不限于违反合同、违反担保、过失、严格责任、虚假陈述以及其他侵权行为。

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: 2020-02-04

目录

前言	9
Informatica 资源	9
Informatica Network	9
Informatica 知识库	9
Informatica 文档	9
Informatica 产品可用性矩阵	9
Informatica Velocity	10
Informatica Marketplace	10
Informatica 全球客户支持部门	10
第 1 章：性能优化概览	11
性能优化概览	11
第 2 章：瓶颈	12
瓶颈概览	12
使用线程统计信息	13
根据线程统计信息消除瓶颈	13
示例	13
目标瓶颈	14
确定目标瓶颈	14
消除目标瓶颈	14
源瓶颈	14
确定源瓶颈	14
消除源瓶颈	15
映射瓶颈	15
确定映射瓶颈	16
消除映射瓶颈	16
会话瓶颈	16
确定会话瓶颈	16
消除会话瓶颈	16
系统瓶颈	16
确定系统瓶颈	16
消除系统瓶颈	17
第 3 章：优化目标	19
优化平面文件目标	19
删除索引和键约束	19
增加数据库检查点时间间隔	20
使用批量加载	20
使用外部加载器	20

最大程度减少死锁.	20
增加数据库网络包大小.	21
优化 Oracle 目标数据库.	21
第 4 章： 优化源.	22
优化查询.	22
使用条件筛选器.	22
增加数据库网络包大小.	23
连接到 Oracle 数据库源.	23
使用 Teradata FastExport.	23
使用 tempdb 联接 Sybase 或 Microsoft SQL Server 表.	23
第 5 章： 优化映射.	24
优化映射概览.	24
优化平面文件源.	24
优化换行缓冲区长度.	24
优化带分隔符的平面文件源.	25
优化 XML 和平面文件源.	25
配置一次性读取.	25
优化传递映射.	25
优化筛选器.	25
优化数据类型转换.	26
优化表达式.	26
提取通用逻辑.	26
将汇总函数调用次数降至最低.	26
使用局部变量替换通用表达式.	27
选择数值而非字符串运算.	27
优化字符-字符与字符-变长字符的比较.	27
选择 DECODE 而非 LOOKUP.	27
使用运算符而非函数.	27
优化 IIF 函数.	27
计算表达式.	28
优化外部过程.	28
第 6 章： 优化转换.	29
优化汇总器转换.	29
按简单列分组.	29
使用已排序输入.	30
使用增量汇总.	30
汇总数据之前对其进行筛选.	30
限制端口连接数.	30
优化自定义转换.	30
优化连接器转换.	30

优化查找转换.	31
使用最优数据库驱动程序.	31
缓存查找表.	31
优化查找条件.	33
筛选查找行.	33
编制查找表的索引.	33
优化多个查找.	33
创建管道查找转换.	33
优化规范器转换.	34
优化序列生成器转换.	34
优化排序器转换.	34
分配内存.	34
分区工作目录.	35
Unicode 模式.	35
优化源限定符转换.	35
优化 SQL 转换.	35
优化 XML 转换.	35
消除转换错误.	36
 第 7 章： 优化会话.	 37
网格.	37
下推优化.	38
并发会话和工作流.	38
缓冲区内存.	38
增加 DTM 缓冲区大小.	38
优化缓冲区块大小.	39
缓存.	39
限制连接的端口数.	39
缓存目录位置.	40
增加缓存大小.	40
使用 64 位版本的 PowerCenter.	40
基于目标的提交.	40
实时处理.	41
刷新延迟.	41
基于源的提交.	41
暂存区域.	41
日志文件.	41
错误跟踪.	41
会话后电子邮件.	42
 第 8 章： 优化网格部署.	 43
优化网格部署概览.	43
存储文件.	43

高带宽共享文件系统文件.	44
低带宽共享文件系统文件.	44
本地存储文件.	44
使用共享文件系统.	44
配置共享文件系统.	44
平衡 CPU 和内存使用情况.	45
配置 PowerCenter 映射和会话.	45
在文件系统之间分发文件.	46
配置会话以分发文件.	46
优化序列生成器转换.	47
 第 9 章： 优化 PowerCenter 组件.	 48
优化 PowerCenter 组件概览.	48
优化 PowerCenter 存储库性能.	48
存储库服务进程和存储库的位置.	48
对对象查询中的条件进行排序.	48
使用单节点 DB2 数据库表空间.	49
优化数据库架构.	49
存储库服务的对象缓存.	49
优化弹性.	50
优化集成服务性能.	50
使用本地和 ODBC 驱动程序.	50
以 ASCII 数据移动模式运行集成服务.	51
为存储库服务缓存 PowerCenter 元数据.	51
 第 10 章： 优化系统.	 52
优化系统概览.	52
提高网络速度.	52
使用多个 CPU.	53
减少分页.	53
使用处理器绑定.	53
 第 11 章： 使用管道分区.	 54
使用管道分区概览.	54
增加分区数目.	54
选择最好的分区类型.	55
使用多个 CPU.	55
为分区优化源数据库.	55
调整数据库.	56
对已排序数据进行分组.	56
优化单排序查询.	56
为分区优化目标数据库.	57

附录 A: 性能计数器.....	58
性能计数器概览.....	58
Errorrows 计数器.....	58
Readfromcache 和 Writetocache 计数器.....	58
Readfromdisk 和 Writetodisk 计数器.....	59
Rowsinlookupcache 计数器.....	59
索引.....	60

前言

要了解运行时瓶颈和如何调整实现最佳性能，请参阅《PowerCenter® 性能优化指南》。

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, 11](#)

性能优化概览

性能优化的目标是通过消除性能瓶颈来提高会话性能。要调整会话性能，首先标识性能瓶颈，消除该瓶颈，然后标识下一个性能瓶颈，直到对会话性能满意为止。调整会话性能时，可以使用测试负载选项来运行会话。

如果调整所有瓶颈，则可以通过增加会话中的管道分区数进一步提高会话性能。添加分区后，可在处理会话时利用更多的系统硬件，从而提高性能。

因为确定提高性能的最佳方法会十分复杂，请一次更改一个变量，并在更改之前和之后记录会话时间。如果会话性能没有提高，您可能需要返回原始配置。

请完成以下任务以提高会话性能：

1. **优化目标。**使集成服务能够有效地写入目标。
2. **优化源。**使集成服务能够有效地读取源数据。
3. **优化映射。**使集成服务能够有效地转换和移动数据。
4. **优化转换。**使集成服务能够有效地处理映射中的转换。
5. **优化会话。**使集成服务能够更快地运行会话。
6. **优化网格部署。**使集成服务能够以最佳性能在网格上运行。
7. **优化 PowerCenter 组件。**使集成服务和存储库服务能够达到最佳运行效果。
8. **优化系统。**使 PowerCenter 服务进程能够更快地运行。

第 2 章

瓶颈

本章包括以下主题：

- [瓶颈概览, 12](#)
- [使用线程统计信息, 13](#)
- [目标瓶颈, 14](#)
- [源瓶颈, 14](#)
- [映射瓶颈, 15](#)
- [会话瓶颈, 16](#)
- [系统瓶颈, 16](#)

瓶颈概览

性能优化的第一步是确定性能瓶颈。性能瓶颈可能出现在源和目标数据库、映射、会话和系统中。方法是标识一个性能瓶颈、消除该瓶颈，然后标识下一个性能瓶颈，直到对性能满意为止。

按以下顺序查找性能瓶颈：

1. 目标
2. 源
3. 映射
4. 会话
5. 系统

使用以下方法可标识性能瓶颈：

- **运行测试会话。**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会话，以标识源和目标瓶颈。
- **分析性能详细信息。**分析性能计数器等性能详细信息，以确定会话性能下降的位置。
- **分析线程统计信息。**分析线程统计信息来确定分区点的最优数量。
- **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、分页以确定系统瓶颈。还可以使用 workflow 监视查看系统资源使用情况。

使用线程统计信息

可以使用会话日志中的线程统计信息来确定源、目标或转换瓶颈。默认情况下，集成服务使用一个读取器线程、一个转换线程和一个写入器线程处理一个会话。繁忙百分比最高的线程就是该会话中的瓶颈。

会话日志提供以下线程统计信息：

- **运行时间。**线程运行的时间量。
- **空闲时间。**线程处于空闲的时间量。它包括线程等待应用程序中其他线程处理的时间。空闲时间包括线程被集成服务阻止的时间，但不包括线程由操作系统阻止的时间。
- **繁忙时间。**由以下公式得出线程繁忙占运行时间的百分比：

$$(\text{run time} - \text{idle time}) / \text{run time} \times 100$$

如果总运行时间短暂，比如低于 60 秒，您可以忽略较高的繁忙百分比。这并不一定表示存在瓶颈。

- **线程工作时间。**集成服务处理一个线程中的每个转换所需的时间的百分比。会话日志显示转换线程工作时间的以下信息：

```
Thread work time breakdown:
  <transformation name>: <number> percent
  <transformation name>: <number> percent
  <transformation name>: <number> percent
```

如果转换需要的时间较少，会话日志不包含该转换。如果因为运行时间较短，线程没有准确的统计信息，会话日志报告统计信息不准确。

根据线程统计信息消除瓶颈

请完成以下任务来根据线程统计信息消除瓶颈：

- 如果读取器或写入器线程处于 100% 繁忙，请考虑在源或目标的端口中使用字符串数据类型。非字符串端口需要更多处理。
- 如果转换线程 100% 繁忙，考虑在段中增加一个分区点。当您向映射添加分区点时，集成服务增加用于会话的转换线程的数目。然而，如果该计算机已在处于或接近满负荷运行，请不要添加更多线程。
- 如果一次转换需要的处理时间高于其他转换，请考虑向转换添加传递分区点。

示例

当您运行会话时，会话日志列出运行信息和线程的统计信息，如以下文本所示：

```
***** RUN INFO FOR TGT LOAD ORDER GROUP [1], CONCURRENT SET [1] *****
Thread [READER_1_1_1] created for [the read stage] of partition point [SQ_two_gig_file_32B_rows] has
completed.
  Total Run Time = [505.871140] secs
  Total Idle Time = [457.038313] secs
  Busy Percentage = [9.653215]
Thread [TRANSF_1_1_1] created for [the transformation stage] of partition point
[SQ_two_gig_file_32B_rows] has completed.
  Total Run Time = [506.230461] secs
  Total Idle Time = [1.390318] secs
  Busy Percentage = [99.725359]
  Thread work time breakdown:
    LKP_ADDRESS: 25.000000 percent
    SRT_ADDRESS: 21.551724 percent
    RTR_ZIP_CODE: 53.448276 percent
Thread [WRITER_1_*_1] created for [the write stage] of partition point [scratch_out_32B] has completed.
  Total Run Time = [507.027212] secs
  Total Idle Time = [384.632435] secs
  Busy Percentage = [24.139686]
```

在此会话日志中，转换线程的总运行时间为 506 秒，繁忙状态百分比为 99.7%。这意味着该转换线程 506 秒内没有空闲。读取器和写入器的繁忙百分比明显较小，大约为 9.6% 和 24%。在此会话中，转换线程是映射中的瓶颈。

要确定转换线程中的哪个转换是瓶颈，在线程工作时间细分中查看每个转换的繁忙百分比。在此会话日志中，转换 RTR_ZIP_CODE 繁忙百分比为 53%。

目标瓶颈

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。检查点时间间隔较小、数据库网络包大小较小或重度加载操作中出现问题都可能导致目标瓶颈。

确定目标瓶颈

要确定目标瓶颈，请完成以下任务：

- 配置会话副本以写入一个平面文件目标。如果会话性能显著提高，则存在目标瓶颈。如果会话已写入平面文件目标，您可能没有遇到目标瓶颈。
- 读取会话日志中的线程统计信息。当集成服务在写入器线程上花费的时间比转换或读取器线程更多，则存在目标瓶颈。

消除目标瓶颈

请完成以下任务来消除目标瓶颈：

- 让数据库管理员通过优化查询来优化数据库性能。
- 增加数据库网络包大小。
- 配置索引和键约束。

相关主题：

- [“优化目标”页面上 19](#)

源瓶颈

当集成服务通过源数据库读取时，可能出现性能瓶颈。低效的查询或较小的数据库网络包大小可能会导致源瓶颈。

确定源瓶颈

您可以在会话日志读取线程统计信息，以确定源是否是瓶颈。当集成服务在读取器线程上花费的时间比转换或写入器线程更多，则存在源瓶颈。

如果会话从关系源中读取数据，请使用以下方法标识源瓶颈：

- 筛选器转换
- 读取测试映射

- 数据库查询

如果会话从平面文件源中读取，可能没有源瓶颈。

使用筛选器转换

您可以在映射中使用筛选器转换，来测量读取源数据所花费的时间。

在每个源限定符后添加筛选器转换。将筛选条件设置为 false，这样就没有通过筛选器转换处理任何数据。如果运行新会话所需的时间仍然差不多，则存在源瓶颈。

使用读取测试映射

您可以创建一个读取测试映射，以确定源瓶颈。通过删除转换中的映射，读取测试映射分离该读取查询。

要创建读取测试映射，请完成以下步骤：

1. 创建原始映射的副本。
2. 在映射副本中，仅保留源、源限定符和任何自定义联接或查询。
3. 删除所有转换。
4. 将源限定符连接到文件目标。

针对读取测试映射运行会话。如果会话性能与原始会话相似，则存在源瓶颈。

使用数据库查询

要确定源瓶颈，请直接对源数据库执行读取查询。

直接从会话日志复制读取查询。使用查询工具（例如 isql）对源数据库执行查询。在 Windows 上，您可以加载文件中的查询结果。在 UNIX 上，您可以加载 /dev/null 中的查询结果。

测量查询执行时间以及查询返回第一行所需的时间。

消除源瓶颈

完成以下任务来消除源瓶颈：

- 如果集成服务从平面文件源读取，设置集成服务每行读取的字节数。
- 让数据库管理员通过优化查询来优化数据库性能。
- 增加数据库网络包大小。
- 配置索引和键约束。
- 如果数据库查询中两次时间度量之间存在较长的延迟，您可以使用优化器提示。

相关主题：

- [“优化源”页面上 22](#)

映射瓶颈

如果您确定不存在源或目标瓶颈，则可能存在映射瓶颈。

确定映射瓶颈

要确定映射瓶颈，请完成以下任务：

- 读取会话日志中的线程统计信息和工作时间统计信息。如果集成服务在转换线程上花费的时间比写入器或读取器线程更多，则存在转换瓶颈。如果集成服务在某个转换上花费更多时间，则此为转换线程中的瓶颈。
- 分析性能计数器。Errorrows 和 rowsinlookupcache 的高计数器指示存在映射瓶颈。
- 在每个目标定义前添加筛选器转换。将筛选条件设置为 False，不会将任何数据加载到目标表。如果运行新会话所需的时间与原始会话相同，则存在映射瓶颈。

消除映射瓶颈

要消除映射瓶颈，请优化映射的转换设置。

相关主题：

- [“优化映射” 页面上 24](#)

会话瓶颈

如果不存在源、目标或映射瓶颈，则可能存在会话瓶颈。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会话瓶颈。

确定会话瓶颈

要确定会话瓶颈，请分析性能详细信息。性能详细信息显示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。

消除会话瓶颈

要消除会话瓶颈，请优化会话。

相关主题：

- [“优化会话” 页面上 37](#)

系统瓶颈

调整源、目标、映射和会话后，应考虑调整系统，以防止系统瓶颈。集成服务使用系统资源处理转换，运行会话，以及读取和写入数据。集成服务还使用系统内存创建转换（汇总器、连接器、查找、排序器、XML 和等级转换）的缓存文件。

确定系统瓶颈

您可以在工作流监视中查看系统资源使用情况。可以使用系统工具监视 Windows 和 UNIX 系统。

使用 workflow 监视确定系统瓶颈

在集成服务上运行任务进程时，可以在 workflow 监视中查看集成服务属性，以了解系统 CPU、内存和交换的使用情况。使用以下集成服务属性来确定性能问题：

- **CPU%.**CPU 使用率百分比包括在系统上运行的其他外部任务。
- **内存使用情况。**内存使用率百分比包括在系统上运行的其他外部任务。如果内存使用率接近 95%，请检查在该系统上运行的任务是否使用了在 workflow 监视中标识的数量，或者是否存在内存泄漏。要排除故障，请在运行会话之前和之后使用系统工具检查内存使用量，然后将此结果与运行会话时的内存使用量进行比较。
- **交换使用情况。**由于可能存在内存泄漏或大量并发任务，交换使用情况是分页的结果。

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详细的信息。

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。

消除系统瓶颈

请完成以下任务以消除系统瓶颈：

- 如果 CPU 使用率超过 80%，请检查并发运行的任务的数目。考虑更改负载或使用网格来将任务分发到不同节点。如果您不能减少负载，请考虑添加更多的处理器。
- 如果发生交换，请增加物理内存或减少磁盘上的内存密集型应用程序数量。
- 如果内存压力过大（抖动），请考虑添加更多物理内存。

- 如果时间百分比较高，请将 PowerCenter 缓存调整为使用内存中的缓存而不是写入磁盘。如果调整缓存，请求仍处于队列中，并且磁盘繁忙百分比超过了 50%，请添加另一个磁盘设备或升级为更快的磁盘设备。您还可以为会话中的每个分区使用一个单独的磁盘。
- 如果物理磁盘队列长度大于 2，请考虑添加另一个磁盘设备或升级该磁盘设备。您还可以为读取器、写入器和转换线程使用单独的磁盘。
- 考虑提高网络带宽。
- 当您调整 UNIX 系统时，为一个主要数据库系统优化服务器。
- 如果等待 I/O (%wio) 花费时间的百分比较高，可以考虑使用其他未充分利用的磁盘。例如，如果源数据、目标数据、查找、等级和汇总缓存文件都位于同一磁盘上，考虑把它们放在不同的磁盘上。

相关主题：

- [“减少分页” 页面上 53](#)
- [“优化系统” 页面上 52](#)

第 3 章

优化目标

本章包括以下主题：

- [优化平面文件目标, 19](#)
- [删除索引和键约束, 19](#)
- [增加数据库检查点时间间隔, 20](#)
- [使用批量加载, 20](#)
- [使用外部加载器, 20](#)
- [最大程度减少死锁, 20](#)
- [增加数据库网络包大小, 21](#)
- [优化 Oracle 目标数据库, 21](#)

优化平面文件目标

如果您为平面文件目标使用共享存储目录，通过确保共享存储目录位于专用于存储和管理文件而不是执行其他任务的计算机上，从而可以优化会话性能。

如果集成服务在单个节点上运行且该会话写入平面文件目标，您可以通过写入集成服务进程节点的本地平面文件目标来优化会话性能。

删除索引和键约束

当您在目标表中定义键约束或索引时，您减缓了对这些表的数据加载。为提高性能，请在运行会话前去除索引和键约束。在会话完成后，可以重建这些索引和键约束。

如果您决定要定期去除并重建索引和键约束，每次运行会话时可以使用以下方法执行这些操作：

- 使用预加载和加载后存储过程。
- 使用会话前和会话后 SQL 命令。

注意：要优化性能，仅在必要时使用基于约束的加载。

增加数据库检查点时间间隔

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。

注意：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。

使用批量加载

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 数据库的会话性能。在会话属性中配置批量加载。

批量加载时，集成服务将绕过数据库日志，这样可以提高性能。然而，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，请权衡提高会话性能的重要性与恢复不完整会话的能力。

当批量加载到 Microsoft SQL Server 或 Oracle 目标时，定义较大的提交间隔，以提高性能。Microsoft SQL Server 和 Oracle 每次提交后启动新的批量加载事务。增加提交时间间隔可以减少批量加载事务次数，从而提高了性能。

相关主题：

- [“基于目标的提交” 页面上 40](#)

使用外部加载器

要提高会话性能，配置 PowerCenter 以便为以下类型目标数据库使用外部加载器：

- IBM DB2 EE 或 EEE
- Oracle

当您使用带有多个分区的管道向 Oracle 数据库加载数据时，如果使用与管道相同数量的分区创建 Oracle 目标表，可以提高性能。

- Sybase IQ

如果 Sybase IQ 数据库在 UNIX 系统上是集成服务进程的本地数据库，可以通过直接从命名管道将数据加载到目标表来提高性能。如果在网络上运行集成服务，配置负载均衡器检查资源、使 Sybase IQ 成为资源并使资源在网格的所有节点上可用。然后，在 Workflow Manager 中，将 Sybase IQ 资源分配到适用的会话。

- Teradata

最大程度减少死锁

如果集成服务试图写入目标时遇到死锁，死锁只会影响同一目标连接组中的目标。集成服务仍写入其他目标连接组中的目标。

遇到死锁会降低会话性能。要提高会话性能，可以提高集成服务用于在会话中写入目标的目标连接组的数量。要为会话中的每个目标使用不同的目标连接组，请为每个目标实例使用不同的数据库连接名称。您可以为每个连接名称指定相同的连接信息。

增加数据库网络包大小

如果您写入 Oracle、Sybase ASE 或 Microsoft SQL Server 目标，则可以增大网络包大小来提高性能。增加网络包大小，以允许更大的包同时通过网络。根据您写入的数据库增加网络包大小：

- **Oracle。**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）。
- **Sybase ASE 和 Microsoft SQL Server。**有关如何增大包大小的信息，请参阅数据库文档。

对于 Sybase ASE 或 Microsoft SQL Server，您还必须在 Workflow Manager 中更改关联连接对象的包大小，以反映数据库服务器包大小。

优化 Oracle 目标数据库

如果目标数据库是 Oracle，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。

当写入 Oracle 数据库时，请检查数据库对象的存储子句。请确保表正在使用较大的初始值和下一个值。数据库还应在单独的表空间中存储表和索引数据，最好在不同的磁盘中。

当写入 Oracle 数据库时，数据库在加载期间使用回滚或撤消段。要求 Oracle 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。

要优化 Oracle 数据库，请调整 Oracle 恢复日志。Oracle 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。您可以查看 init.ora 文件中的恢复日志属性。

如果集成服务在单个节点上运行，并且 Oracle 实例为集成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。

有关优化 Oracle 数据库的详细信息，请参阅 Oracle 文档。

第 4 章

优化源

本章包括以下主题：

- [优化查询, 22](#)
- [使用条件筛选器, 22](#)
- [增加数据库网络包大小, 23](#)
- [连接到 Oracle 数据库源, 23](#)
- [使用 Teradata FastExport, 23](#)
- [使用 tempdb 联接 Sybase 或 Microsoft SQL Server 表, 23](#)

优化查询

如果一个会话联接一个源限定符中的多个源表，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。

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。

集成服务用于读取数据的查询在会话日志中显示。还可以在源限定符转换中找到该查询。请数据库管理员分析该查询，然后为源表创建优化器提示和索引。

如果查询开始执行的时间与 PowerCenter 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。

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。

使用条件筛选器

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。可以在源限定符中使用 PowerCenter 条件筛选器来提高性能。

是否应使用 PowerCenter 条件筛选器来提高性能取决于会话。例如，如果多个会话同时从相同源中读取数据，PowerCenter 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，某些会话执行的速度可能会更快。您可以同时使用数据库筛选器和 PowerCenter 筛选器对会话进行测试，以确定哪种方法能够提高性能。

增加数据库网络包大小

如果您从 Oracle、Sybase ASE 或 Microsoft SQL Server 源读取数据，则可以增大网络包大小来提高性能。增加网络包大小，以允许更大的包同时通过网络。根据您从中读取数据的数据库增加网络包大小：

- **Oracle。**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）。
- **Sybase ASE 和 Microsoft SQL Server。**有关如何增大包大小的信息，请参阅数据库文档。

对于 Sybase ASE 或 Microsoft SQL Server，您还必须在 Workflow Manager 中更改关联连接对象的包大小，以反映数据库服务器包大小。

连接到 Oracle 数据库源

如果在单个节点上运行集成服务，并且 Oracle 实例为集成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。

使用 Teradata FastExport

FastExport 是一个实用工具，它使用多个 Teradata 会话来快速从 Teradata 数据库导出大量数据。您可以创建一个使用 FastExport 来快速阅读 Teradata 源的 PowerCenter 会话。要使用 FastExport，使用 Teradata 源数据库创建一个映射。在会话中，使用 FastExport 读取器而不是关系读取器。使用连接到希望在会话中导出的 Teradata 表的 FastExport 连接。

使用 tempdb 联接 Sybase 或 Microsoft SQL Server 表

当连接 Sybase 或 Microsoft SQL Server 数据库上的大型表时，有可能通过作为内存中数据库创建 tempdb 来分配足够的内存，从而提高性能。有关详细信息，请参阅 Sybase 或 Microsoft SQL Server 文档。

第 5 章

优化映射

本章包括以下主题：

- [优化映射概览, 24](#)
- [优化平面文件源, 24](#)
- [配置一次性读取, 25](#)
- [优化传递映射, 25](#)
- [优化筛选器, 25](#)
- [优化数据类型转换, 26](#)
- [优化表达式, 26](#)
- [优化外部过程, 28](#)

优化映射概览

映射级别优化可能需要一段时间才能实现，但可以显著提高会话性能。优化目标和源后，请重点执行映射级别优化。

通常，可以通过减少映射中的转换数量以及删除转换之间不必要的链接来优化映射。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可将移动的数据量降至最低。

优化平面文件源

请完成以下任务以优化平面文件源：

- 优化换行缓冲区长度。
- 优化带分隔符的平面文件源。
- 优化 XML 和平面文件源。

优化换行缓冲区长度

如果会话从平面文件源中读取数据，您可以通过设置集成服务每行读取的字节数来提高会话性能。默认情况下，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，您可以在会话属性中缩短换行缓冲区长度。

优化带分隔符的平面文件源

如果源为带分隔符的平面文件，您必须指定分隔符字符以分隔源文件中的数据列。此外还必须指定转义符。如果在分隔符字符之前包含转义符，集成服务会将分隔符字符作为常规字符进行读取。如果源平面文件不包含引号或转义字符，则可以提高会话性能。

优化 XML 和平面文件源

XML 文件通常由于存在标记信息而大于平面文件。XML 文件的大小取决于 XML 文件中的标记级别。标记越多，文件大小越大。因此，集成服务可能需要更长的时间来读取和缓存 XML 源。

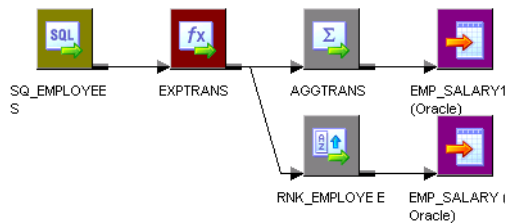
配置一次性读取

一次性读取允许您使用一个源限定符填充多个目标。如果您的多个会话使用相同的源，请考虑使用一次性读取。您可以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源限定符。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和会话中，则可以强制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，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。

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。

下图显示了一次性读取，在该过程中，映射将在表达式转换之后拆分：



优化传递映射

您可以优化传递映射的性能。要直接从源传递到目标而无需任何其他转换，请将源限定符转换直接连接到目标。如果使用入门向导来创建传递映射，该向导将在源限定符转换和目标之间创建表达式转换。

优化筛选器

请使用以下转换之一筛选数据：

- **源限定符转换。**源限定符转换从关系源中筛选行。

- **筛选器转换。**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。

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。在源限定符转换中使用筛选器以在源中删除行。源限定符转换限制从关系源中提取的行集。

如果不能在源限定符转换中使用筛选器，请使用筛选器转换并将其移至尽可能靠近该源限定符转换的位置，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。

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。

注意：如果您不希望保留被拒绝的行，还可以使用筛选器或路由器转换以从更新策略转换中删除被拒绝的行。

优化数据类型转换

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。

请使用以下数据类型转换来提高系统性能：

- **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“字符”或“变长字符”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会话性能。**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。

优化表达式

还可以优化转换中使用的表达式。如果可能，请隔离执行速度缓慢的表达式并将其简化。

请完成以下任务以隔离执行速度缓慢的表达式：

1. 逐一从映射中删除表达式。
2. 运行映射以确定运行不包含转换的映射所需的时间。

如果会话运行时间差别较大，请寻找优化执行速度缓慢的表达式的方法。

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，集成服务都必须搜索数据并对其进行分组。例如，在以下表达式中，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，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。

选择数值而非字符串运算

与处理字符串操作相比，集成服务处理数值操作的速度更快。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。

优化字符-字符与字符-变长字符的比较

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 中配置集成服务以便数据集成服务不裁减“字符”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，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，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,
))))))

```

此表达式需要 8 个 IIF、16 个 AND 和至少 24 个比较。

如果利用 IIF 函数，则可以将该表达式重新编写为：

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

其结果是得到三个 IIF、两个比较、两个相加和一个更快的会话。

计算表达式

如果不能确定哪些表达式会降低性能，可以评估表达式性能以确定问题。

要计算表达式性能，请完成以下步骤：

1. 确定包含原始表达式的会话的运行时间。
2. 复制该映射并将一半复杂表达式替换为常数。
3. 运行编辑后的会话并确定运行时间。
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. 运行编辑后的会话并确定运行时间。

优化外部过程

如果外部过程需要从输入组中交替读取，您可能需要阻止输入数据。如果没有阻止功能，您需要编写过程代码以缓冲传入数据。您可以阻止而非缓存输入数据，这样通常可以提高会话性能。

例如，您需要创建包含两个输入组的外部过程。该外部过程从第一个输入组中读取行，然后从第二个输入组中读取行。如果使用阻止功能，您可以编写外部过程代码，以便在它处理一个输入组中的数据时阻止其他输入组中的数据。当您编写外部过程代码以阻止数据时，可以提高性能，因为该过程不需要将源数据复制到缓冲区。但是，您编写的外部过程可能会分配缓冲区，并将一个输入组中的数据复制到该缓冲区，直到它准备好处理数据。将源数据复制到缓冲区会降低性能。

第 6 章

优化转换

本章包括以下主题：

- [优化汇总器转换, 29](#)
- [优化自定义转换, 30](#)
- [优化联接器转换, 30](#)
- [优化查找转换, 31](#)
- [优化规范器转换, 34](#)
- [优化序列生成器转换, 34](#)
- [优化排序器转换, 34](#)
- [优化源限定符转换, 35](#)
- [优化 SQL 转换, 35](#)
- [优化 XML 转换, 35](#)
- [消除转换错误, 36](#)

优化汇总器转换

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。

使用以下准则来优化汇总器转换的性能：

- 按简单列分组。
- 使用已排序输入。
- 使用增量汇总。
- 汇总数据之前，请对其进行筛选。
- 限制端口连接数。

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

要提高会话性能，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”选项可减少对汇总缓存的使用。使用“已排序输入”选项时，集成服务假定所有数据都按组排序。当集成服务读取组的行时，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”选项可降低会话过程中缓存的数据量，提高性能。将此选项与源限定符“已排序端口数”选项或排序器转换搭配使用，可将已排序数据传递到汇总器转换。

在包含多个分区的会话中使用“已排序输入”选项时，可以提高性能。

使用增量汇总

如果您能捕捉来自源的影响程度不到目标一半的更改，则可以使用增量汇总来优化汇总器转换的性能。

当您使用增量汇总时，会将源中捕捉的更改应用到会话中的汇总计算。每次运行会话时，集成服务将以增量方式更新目标，而不是处理整个源并重新计算同样的计算。

您可以增加索引和数据缓存大小，以容纳所有数据而不需要分页到磁盘。

相关主题：

- [“增加缓存大小”页面上 40](#)

汇总数据之前对其进行筛选

在汇总数据之前对其进行筛选。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减少汇总器转换在数据缓存中存储的数据量。

优化自定义转换

集成服务可以将单个行传递到自定义转换过程或数组中的行块。您可以编写过程代码以指定过程接收一行还是行块。

过程接收行块时，可以提高性能：

- 可以降低集成服务和过程执行的函数调用数量。集成服务调用输入行通知函数的次数变少，过程调用输出通知函数的次数也变少。
- 可以增加数据的内存访问区域空间。
- 可以编写过程代码以针对数据块而非每行数据执行算法。

优化连接器转换

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。可以查看连接器性能计数器信息以确定是否需要优化连接器转换。

请参考以下提示来提高联接器转换的会话性能：

- **指定主源作为具有较少重复键值的源。**处理已排序联接器转换时，集成服务一次缓存一百个唯一键对应的行。如果主源中包含许多具有相同键值的行，集成服务必须缓存更多行，并且性能会降低。
- **指定主源作为具有较少行的源。**会话期间，联接器转换会将详细信息源的每个行与主源进行比较。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。**与在会话中执行联接相比，在数据库中执行联接的速度更快。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，产生的行数较少。在某些情况下，您无法在数据库中执行联接，例如联接来自两个不同数据库或平面文件系统的表。

要在数据库中执行联接，请使用以下选项：

- 创建会话前存储过程以联接数据库中的表。
- 使用源限定符转换执行联接。
- **尽可能联接已排序数据。**要提高会话性能，请将联接器转换配置为使用已排序输入。将联接器转换配置为使用已排序数据时，集成服务会通过将磁盘输入和输出降至最低来提高性能。处理大型数据集时，将会获得最显著的性能提高。对于未排序联接器转换，请将具有较少行的源指定为主源。

优化查找转换

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

如果使用查找转换，请执行以下任务来提高性能：

- 使用最优数据库驱动程序。
- 缓存查找表。
- 优化查找条件。
- 筛选查找行。
- 编制查找表的索引。
- 优化多个查找。
- 创建管道查找转换并在生成查找源的管道中配置分区。

使用最优数据库驱动程序

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会话性能优于 ODBC 驱动程序。

缓存查找表

如果映射中包含查找转换，您可能需要启用查找缓存。启用缓存时，集成服务会在会话过程中缓存查找表并查询查找缓存。如果未启用此选项，集成服务将逐行查询查找表。

无论是否缓存查找表，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会话性能。一般来说，您需要缓存小于 300 MB 的查找表。

完成以下任务以进一步提高查找转换的性能：

- 使用合适的缓存类型。
- 启用并发缓存。

- 优化查找条件匹配。
- 减少已缓存的行数。
- 替代 ORDER BY 语句。
- 使用具有更大内存的计算机。

相关主题：

- [“缓存” 页面上 39](#)

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。如果确定查找表在会话运行之间保持不变，可使用此功能。使用持久性缓存可以提高性能，因为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会话时，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，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，集成服务会同时构建缓存，而不是按顺序构建。会话中包含大量可能需要一段时间才能完成的活动转换时（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，集成服务不再等待活动会话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。

替代 ORDER BY 语句

默认情况下，集成服务将生成已缓存查找的 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。

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

要提高会话性能，请在具有大量内存的集成服务节点上运行会话。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 (!=)

筛选查找行

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

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。会话日志包含 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。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，即使启用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。

要确定哪些查找转换处理的数据最多，请检查每个查找转换的 Lookup_rowsinlookupcache 计数器。调整查找表达式后，查找转换在此计数器中的数字可能会更大。如果这些表达式可以得到优化，会话性能将得到提高。

相关主题：

- [“优化表达式”页面上 26](#)

创建管道查找转换

包含管道查找转换的映射包括部分管道，其中包含查找源和源限定符。集成服务会处理该管道中的查找源数据。它会将查找源数据传递给包含查找转换的管道，并创建缓存。

该部分管道是会话属性中单独的目标加载顺序组。您可以在该管道中配置多个分区以提高性能。

优化规范器转换

规范器转换可生成行。要优化性能，请将规范器转换置于尽可能接近目标的位置。

优化序列生成器转换

要优化序列生成器转换，请创建可重用的序列生成器并同时在多个映射中使用它。此外，配置“缓存值数”属性。

“缓存值数”属性可确定集成服务一次可缓存的值数量。请确保“缓存值数”不会过小。请考虑将“缓存值数”配置为大于 1000 的值。

如果不需要缓存值，请将“缓存值数”设置为 0。不使用缓存的序列生成器转换比需要缓存的转换速度更快。

如果连接序列生成器转换中的 CURRVAL 端口，则集成服务将在每个块中处理一行。可以仅连接映射中的 NEXTVAL 端口以优化性能。

相关主题：

- [“优化序列生成器转换”页面上 47](#)

优化排序器转换

完成以下任务以优化排序器转换：

- 分配足够的内存以对数据进行排序。
- 为排序器转换中的每个分区指定不同的工作目录。
- 在 ASCII 模式下运行 PowerCenter 集成服务。

分配内存

要获得最佳性能，请使用小于或等于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。如果集成服务无法分配足够的内存以对数据进行排序，会话将失败。

如果传入数据量大于排序器缓存大小量，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，集成服务所需的磁盘空间至少为传入数据量的两倍。如果传入数据量远大于排序器缓存大小，集成服务所需的磁盘空间可能不止是工作目录可用的磁盘空间量的两倍。

注意：会话日志包含输入行计数和排序器转换的传入数据的大小。

例如，当集成服务处理排序器转换时，将显示以下消息：

`SORT_40422 排序器转换 [srt_1_Billion_FF] 输出的结尾。已处理 999999999 行(866325637228 个输入字节；868929593344 个临时 I/O 字节)`

在该消息中，输入行数为 999999999，输入行大小为 866325637228。

分区工作目录

集成服务在对数据进行排序时创建临时文件。它将它们存储在工作目录中。您可以将集成服务节点上的任意目录指定为工作目录。默认情况下，集成服务使用为 \$PMTempDir 服务进程变量指定的值。

对包含排序器转换的会话进行分区时，可以为管道中的每个分区指定不同的工作目录。要提高会话性能，请在集成服务节点上的物理分隔磁盘上指定工作目录。

Unicode 模式

要优排序器转换，请在 ASCII 模式下运行 PowerCenter 集成服务。如果 PowerCenter 集成服务在 Unicode 模式下运行，排序器转换会将额外数据溢出到磁盘。

优化源限定符转换

如果您希望集成服务从源中选择唯一值，请对源限定符转换使用“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。

优化 SQL 转换

创建 SQL 转换时，可以将该转换配置为使用外部 SQL 查询或在转换中定义的查询。如果将 SQL 转换配置为在脚本模式下运行，集成服务将处理每个输入行的外部 SQL 脚本。当转换在查询模式下运行时，集成服务将处理您在转换中定义的 SQL 查询。

集成服务每次处理会话中的新查询时，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发生更改时，可能会降低性能。

当转换在查询模式下运行时，在转换中构造静态查询可提高性能。静态查询语句不会更改，但查询子句中的数据会更改。要创建静态查询，请在 SQL 编辑器中使用参数绑定，而非字符串置换。使用参数绑定时，可以将查询子句中的参数设置为转换输入端口中的值。

SQL 查询中包含提交和回滚查询语句时，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。

创建 SQL 转换时，可以配置转换连接到数据库的方式。您可以选择静态连接或者在运行时将连接信息传递到转换。

将转换配置为使用静态连接时，可以从 Workflow Manager 连接中选择一个连接。在会话过程中，SQL 转换连接到数据库一次。当您传递动态连接信息时，SQL 转换将在每次转换处理输入行时连接到数据库。

优化 XML 转换

您可以删除 XML 解析器转换中未计划的组以及从 XML 源定义中删除未计划的组。您不需要为这些未计划的组分配内存，但是需要维护主键-外键约束。

消除转换错误

如果存在大量转换错误，则会降低集成服务的性能。每遇到一个转换错误，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。接下来，集成服务通常会将行写入会话日志文件。

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。检查会话日志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。

如果未设置错误阈值，集成服务会继续处理错误行，从而增加会话运行时间。要优化性能，请将错误阈值设置为在一定的行错误数后停止会话。

如果您需要运行生成大量转换错误的会话，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。

相关主题：

- [“错误跟踪” 页面上 41](#)

第 7 章

优化会话

本章包括以下主题：

- [网格, 37](#)
- [下推优化, 38](#)
- [并发会话和工作流, 38](#)
- [缓冲区内存, 38](#)
- [缓存, 39](#)
- [基于目标的提交, 40](#)
- [实时处理, 41](#)
- [暂存区域, 41](#)
- [日志文件, 41](#)
- [错误跟踪, 41](#)
- [会话后电子邮件, 42](#)

网格

可以使用网格提高会话和工作流性能。网格是分配给一组节点的别名，该节点组使您能够将工作流和会话自动分布到各个节点中。

负载均衡器可以向节点分布任务而不会使任何节点过载。

使用网格时，集成服务会将工作流任务和会话线程分布在多个节点中。负载均衡器可以向节点分布任务而不会使任何节点过载。在网格的节点上运行工作流和会话可实现以下性能改善：

- 平衡集成服务工作负载。
- 加快并发会话处理速度。
- 加快分区处理速度。

集成服务需要 CPU 资源来解析输入数据和设置输出数据的格式。当您在会话的提取和加载步骤中遇到性能瓶颈时，网格可以提高性能。

当内存或临时存储成为性能瓶颈时，网格可以提高性能。PowerCenter 映射包含具有高速缓存的转换时，为每个缓存实例部署足够的内存和单独的磁盘存储可以提高性能。

在网格上运行会话可以提高吞吐量，因为网格提供更多的资源来运行会话。同时在网格上运行较少会话时，性能将会提高。如果并发会话分区的数量小于节点数量，在网格上运行会话比在网格上运行工作流效率更高。

在网格上运行多个会话时，会话子任务会与其他并发会话的子任务共享节点资源。在网格上运行会话需要在不同节点上运行的进程之间相互协调。对于某些映射，在网格上运行会话需要额外的系统开销，以将数据从一个节点移动到另一个节点。除了在每个节点上加载内存和 CPU 资源外，在网格上运行多个会话也会增加网络流量。

在网格上运行工作流时，集成服务将在节点上加载内存和 CPU 资源，而不需要在节点之间的协调。

相关主题：

- [“优化网格部署” 页面上 43](#)

下推优化

要提高会话性能，请将转换逻辑推送到源或目标数据库。根据映射和会话配置，集成服务将对源或目标数据库执行 SQL，而非处理集成服务内的转换逻辑。

并发会话和工作流

如果可能，请并发运行会话和工作流，以提高性能。例如，如果将数据加载到具有维度表和事实表的分析架构，请同时加载维度。

缓冲区内存

集成服务初始化会话时，会分配内存块以保存源和目标数据。集成服务为每个源和目标分区分配至少两个内存块。使用大量源和目标的会话可能需要更多的内存块。如果集成服务无法分配足够的内存块以保存数据，会话将失败。

您可以配置缓冲区内存量，或者可以将集成服务配置为在运行时计算缓冲区设置。

要增加可用内存块的数量，请调整以下会话属性：

- **DTM 缓冲区大小。**在会话属性中的“属性”选项卡上增加 DTM 缓冲区大小。
- **默认缓冲区块大小。**在会话属性中的“配置对象”选项卡上减少缓冲区块大小。

注意：如果启用了数据分区，DTM 缓冲区大小将是分配给所有分区的所有内存缓冲池的总大小。对于包含 n 个分区的会话，请将“DTM 缓冲区大小”至少设置为具有一个分区的会话的值的 n 倍。

增加 DTM 缓冲区大小

DTM 缓冲区大小设置指定集成服务用作 DTM 缓冲区内存的内存量。增加 DTM 缓冲区内存时，集成服务将创建更多缓冲区块，从而在瞬时速度减缓期间提高性能。

增加 DTM 缓冲区内存分配通常会导致性能先提高而后趋平。如果性能未显著提高，则 DTM 缓冲区内存分配不是影响会话性能的因素。

要增加 DTM 缓冲区大小，请打开会话属性，然后单击“属性”选项卡。在“性能”设置中编辑“DTM 缓冲区大小”属性。以缓冲区块大小的倍数增加“DTM 缓冲区大小”属性。

优化缓冲区块大小

如果计算机物理内存有限，且会话中的映射包含大量的源、目标或分区，您可能需要减少缓冲区块大小。

如果您正在操作非常大的数据行，增加缓冲区块大小可以提高性能。如果您不知道行的近似大小，请通过完成以下步骤确定行的尺寸。

评估所需的缓冲区块大小：

1. 在 Mapping Designer 中，打开会话的映射。
2. 打开目标实例。
3. 单击“端口”选项卡。
4. 在目标中添加所有列的精度。
5. 如果映射中有多个目标，请为每个附加目标重复步骤 2 至 4，以计算每个目标的精度。
6. 为映射中的每个源定义重复步骤 2 至 5。
7. 选择所有源和目标精度中的最大精度作为缓冲区块大小计算的总精度。

总精度表示移动最大的数据行所需的总字节数。例如，如果总精度等于 33,000，则集成服务需要缓冲区块中的 33,000 个字节来移动该行。如果缓冲区块大小只有 64,000 个字节，则集成服务一次最多只能移动一行。

要设置缓冲区块大小，请打开会话属性并单击“配置对象”选项卡。在“高级”设置中编辑“默认缓冲区块大小”属性。

正如 DTM 缓冲区内内存分配一样，增加缓冲区块大小可提高性能。如果性能未提高，则缓冲区块大小不是会话性能的一个因素。

缓存

集成服务将索引和数据缓存用于 XML 目标以及汇总器、等级、查找和联接器转换。集成服务会将转换后的数据存储在数据缓存中，然后再将其返回到管道。它将组信息存储在索引缓存中。此外，集成服务使用缓存来存储排序器转换的数据。

要配置高速缓存量，请使用缓存计算器或指定缓存大小。还可以将集成服务配置为在运行时计算高速缓存设置。

如果分配的缓存不足以存储数据，集成服务会在处理会话数据时将数据存储在临时磁盘文件（缓存文件）中。每次集成服务分页到临时文件时，性能都会降低。检查性能计数器，以确定集成服务分页到文件的频率。

请执行以下任务以优化缓存：

- 限制连接的输入/输出端口和仅输出端口的数量。
- 选择最佳缓存目录位置。
- 增加缓存大小。
- 使用 64 位版本的 PowerCenter 运行大型缓存会话。

限制连接的端口数

针对使用数据缓存的转换，限制连接的输入/输出端口和仅输出端口的数量。限制连接的输入/输出端口或输出端口的数量可降低转换在数据缓存中存储的数据量。

缓存目录位置

如果在网格上运行集成服务，并且只有部分集成服务节点能够快速访问共享缓存文件目录，请将每个具有大型缓存的会话配置为在能够快速访问目录的节点上运行。要将会话配置为在能够快速访问目录的节点上运行，请完成以下步骤：

1. 创建 PowerCenter 资源。
2. 使资源可用于能够快速访问目录的节点。
3. 将资源分配给会话。

如果网格中的所有集成服务进程对缓存文件的访问都十分缓慢，请为每个集成服务进程设置单独的本地缓存文件目录。如果集成服务进程在包含缓存目录的同一台计算机上运行，它可以更快地访问缓存文件。

注意：将大量数据缓存到映射或装载的驱动器上时，性能可能会下降。

增加缓存大小

可以配置缓存大小，以指定为处理转换而分配的内存量。配置的内存量取决于要使用的内存缓存和磁盘缓存。如果您配置的缓存大小不足以在内存中处理转换，集成服务将在内存中处理部分转换，并将信息分页到缓存文件，以处理其余的转换。每次集成服务分页到缓存文件时，性能都会降低。

您可以检查会话的性能详细信息以确定集成服务分页到缓存文件的时间。任何汇总器、等级或联接器转换的 *Transformation_readfromdisk* 或 *Transformation_writetodisk* 计数器指示集成服务分页到磁盘以处理转换的次数。

如果会话包含使用缓存的转换，并且您在具有足够内存的计算机上运行该会话，请增加缓存大小以在内存中处理转换。

使用 64 位版本的 PowerCenter

如果处理大量数据或执行内存密集型转换，可以使用 64 位 PowerCenter 版本来提高会话性能。64 位版本提供了更大的内存空间，可以大大减少或消除磁盘输入/输出。

这样可以在以下方面提高会话性能：

- **缓存。**凭借 64 位平台，集成服务便不会局限于 32 位平台的 2 GB 缓存限制。
- **数据吞吐量。**凭借更大的可用内存空间，读取器、写入器和 DTM 线程可以处理更大的数据块。

基于目标的提交

提交时间间隔设置可确定集成服务将数据提交至目标的时间点。集成服务每次提交时，性能都会降低。因此，提交时间间隔越小，集成服务写入目标数据库的频率就会越高，整体性能也就越低。

如果增加提交时间间隔，集成服务提交的次数就会减少，性能就会提高。

增加提交时间间隔时，请考虑目标数据库中的日志文件限制。如果提交时间间隔太大，集成服务可能会填充数据库日志文件，导致会话失败。

因此，请权衡增加提交时间间隔所带来的好处与恢复失败的会话所花费的额外时间。

单击会话属性中的“常规选项”设置以查看并调整提交时间间隔。

实时处理

刷新延迟

刷新延迟可确定集成服务刷新来自源的实时数据的频率。设置的刷新延迟时间间隔越低，集成服务将消息提交至目标的频率就越高。集成服务每次将消息提交至目标时，会话都会消耗更多资源且吞吐量会下降。

增加刷新延迟可提高吞吐量。将刷新延迟增加到某一阈值的过程中，吞吐量将增加，具体取决于硬件和可用资源。

基于源的提交

基于源的提交时间间隔可确定集成服务将实时数据提交至目标的频率。要获得最快的延迟，请将基于源的提交设置为 1。

暂存区域

当您使用暂存区域时，集成服务将对数据执行多个传递。如果可能，请删除暂存区域以提高性能。集成服务可一次性读取多个源，从而减轻对暂存区域的需要。

相关主题：

- [“配置一次性读取” 页面上 25](#)

日志文件

如果不将工作流配置为写入会话和工作流日志文件，工作流的运行速度将更快。工作流和会话始终创建二进制日志。当您将会话或工作流配置写入日志文件时，集成服务将写入日志记录事件两次。可以在 Administrator 工具中访问二进制日志会话和工作流日志。

错误跟踪

要提高性能，请减少集成服务在运行会话时生成的日志事件数量。如果会话中包含大量转换错误，而您不需要进行更正，请将会话跟踪级别设置为“简洁”。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”，则可能会在运行会话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。

会话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。

会话后电子邮件

当您将会话日志附加到会话后电子邮件时，请启用平面文件日志记录。如果启用平面文件日志记录，集成服务将从磁盘中获取会话日志文件。如果不启用平面文件日志记录，集成服务将从日志管理器中获取日志事件，并生成要附加到电子邮件的会话日志文件。当集成服务从日志服务中检索会话日志时，工作流性能会降低，尤其是当会话日志文件较大且日志服务在与主 DTM 不同的节点上运行时。要实现最优性能，请将会话配置为在您配置要附加会话日志的会话后电子邮件时写入日志文件。

第 8 章

优化网格部署

本章包括以下主题：

- [优化网格部署概览, 43](#)
- [存储文件, 43](#)
- [使用共享文件系统, 44](#)
- [在文件系统之间分发文件, 46](#)
- [优化序列生成器转换, 47](#)

优化网格部署概览

当您在网络上运行 PowerCenter 时，您可以配置网格、会话和工作流，从而有效使用资源并实现最大限度可扩展性。

要提高网格上的 PowerCenter 性能，请完成以下任务：

- 将节点添加到网格中。
- 增加存储容量和带宽。
- 使用共享文件系统。
- 当您完成以下任务时，请使用高吞吐量的网络：
 - 通过网络访问源和目标。
 - 使用“网格上的会话”选项时在网格的节点之间传输数据。

存储文件

当配置 PowerCenter 以在网络上运行时，为不同类型的会话文件（如源文件、日志文件和缓存文件）指定存储位置。要提高性能，请将文件存储在最佳位置。例如，在高带宽共享文件系统上存储持久性缓存文件。不同类型文件具有不同存储要求。

可以将文件存储在以下类型的位置：

- **共享文件系统。** 将文件存储在一个共享文件系统上，以使所有集成服务进程可以访问相同的文件。您可以在低带宽和高带宽共享文件系统上存储文件。
- **本地。** 如果其他集成服务进程不需要访问文件，将文件存储在运行集成服务进程的本地计算机上。

高带宽共享文件系统文件

因为可能会在会话期间经常访问以下文件，将这些文件放在高带宽共享文件系统上：

- 源文件，包括用于查找的平面文件。
- 目标文件，包括已分区会话的合并文件。
- 用于查找或增量汇总的持久性缓存文件。
- 网格上的仅启用网格的会话的非持久性缓存文件。

这允许集成服务只构建一次缓存。如果这些缓存文件存储在本地文件系统上，集成服务为每个分区组构建缓存。

低带宽共享文件系统文件

因为不经常在会话期间访问以下文件，将这些文件存储在低带宽共享文件系统上：

- 参数文件或其他配置相关文件。
- 间接源文件或目标文件。
- 日志文件。

本地存储文件

为避免使用共享文件系统时不必要的文件共享，请将以下文件存储在本地：

- 没有为网格启用的会话的非持久性缓存文件，包括排序器转换临时文件。
- 为已分区会话执行连续合并时，用于不同分区的单个目标文件。
- 会话运行结束时删除的其他临时文件。一般情况下，要建立此文件，请为本地文件系统配置 `$PmTempFileDir`。

避免在共享文件系统上存储这些文件，即使带宽较高。

使用共享文件系统

您可以将以下共享文件系统用于文件共享：

- 网络文件系统，如用于 Windows 的 CIFS (SMB) 或用于 UNIX 的网络文件系统 (NFS)。虽然网络文件系统并非用于高性能计算，但是它们仍然适合连续文件访问。
- 群集文件系统。群集文件系统提供了具有高带宽文件访问的一组节点，以及文件和目录的统一命名空间。群集文件系统性能类似于直接附加的本地文件系统。

注意：如果您有“高可用性”选项，请使用群集文件系统。

正确配置和调整对于小型网格性能至关重要。您还可以配置映射和会话，以避免共享文件系统的内在限制。

配置共享文件系统

使用以下一般准则配置共享文件系统：

- 请确保网络具有足够带宽。
- 请确保基本存储具有足够 I/O 带宽。

- 配置共享文件系统后台程序（特别是客户端）以获得足够线程来快速访问文件。例如，IBM 建议您估计需要同时访问的文件数，并为每个文件提供至少两个 biod 线程。

在使用平面文件源或目标的网格上运行并发会话时，请提供足够的线程，以使每个分区可以同时访问所需的源或目标文件。

- 根据访问要求配置共享文件系统的装载点。在使用平面文件源或目标的网格上运行连续会话时，避免可能降低默认预读或后写进程效果的任何配置。文件系统通过预读和后写优化连续文件访问。
- 如有必要，调整共享文件系统预读和后写设置。
- 查看客户端和服务器的共享文件系统缓存设置。增加默认设置可能会提高性能。
- 配置文件系统后释放设置可以在访问数据后释放内存页面。否则，读取或写入大型文件时，系统性能可能会降低。
- 由于访问模式不同，您可以为源和目标以及持久性缓存使用不同装载点。

有关详细信息，请参阅共享文件系统文档。

平衡 CPU 和内存使用情况

不同于本地文件系统，共享文件系统服务器可以使用额外的 CPU 周期访问文件。如果将一个计算节点用作其余节点的共享文件系统服务器，它可能会过载，并成为整个网格的瓶颈。当共享文件系统服务器过载时，CPU 周期可能增加，同时出现重复传输和请求超时。

要避免此问题，使用一个或多个计算机作为 PowerCenter 网格节点的专用共享文件系统服务器。每台计算机应针对所需任务拥有足够的存储、CPU 和网络带宽。

或者，您可以交叉装载共享文件系统服务器，以在网格的节点之间分发文件服务器加载。如果将 PowerCenter 映射和会话配置为使用均衡 I/O 和 CPU 使用量时，交叉装载共享文件系统服务器可以优化性能。如果网格中节点数目较少，并且您具有均衡混合 I/O 和 CPU 使用量，您可能不需要专用共享文件系统服务器。

如果使用多个专用或交叉装载共享文件系统服务器，请尝试在服务器间分发共享文件。

相关主题：

- [“在文件系统之间分发文件” 页面上 46](#)

配置 PowerCenter 映射和会话

提高性能的最重要途径之一是避免不必要的文件共享。正确配置时，共享文件系统可以为源文件和目标文件的连续访问提供良好性能。然而，持久性缓存文件（特别是大型持久性缓存文件）所需的随机访问可能导致更多问题。

要为具有共享文件系统的网格配置持久性缓存文件（如持久性动态查找），请使用以下准则：

- 如有可能，配置会话缓存大小以在内存中保存较小的持久性缓存文件。
- 将排序器转换添加到映射以在持久性查找之前对输入行进行排序。因为排序器转换可以使用本地文件系统，将持久性查找的工作转移到排序器转换可以提高性能。
- 对需要访问查找缓存同一页面的行进行分组，以尽量减少集成服务读取缓存中的每页的次数。
- 当输入数据的大小很大时，使用基于源的提交来管理输入数据，从而允许在内存中进行排序。

例如，您有 4 GB 持久性动态查找，如果不更改映射逻辑则不能减少，且有 10 GB 源数据。首先添加排序器转换对输入数据进行排序，以减少查找缓存的随机访问，然后完成以下任务：

- 配置会话来使用 1 GB 提交时间间隔执行基于源的提交。
- 将排序器转换事务范围设置为“事务”。
- 为 1 GB 缓存大小配置排序器转换，足够用于源输入。

使用此配置，集成服务一次对 1 GB 输入数据进行排序，将行传递给需要访问缓存中的类似数据的持久性查找。

- 如果多个文件系统可用，为每个分区配置缓存文件，以使用不同文件系统。
- 如果多个文件系统可用，配置会话从而将文件分发到不同的文件系统。

在文件系统之间分发文件

假设每个文件系统使用一个独立的物理磁盘子系统，将文件分发到不同的文件系统可以使用文件系统的组合带宽。在使用共享文件系统或对称多处理 (SMP) 的网格上，将文件分发到不同的文件系统可以提高性能。

为优化的 I/O 带宽选择在多个存储设备之间分发文件的文件系统。如果您使用群集文件系统，将在服务器之间分发文件。如果可能，将源、目标和缓存文件放在不同的存储设备上。

当您在文件系统上分发文件时，请使用以下准则：

- **源文件。**如果您将源文件放在某个文件系统上，而该系统启用了集成服务以通过大量文件读取数据，请在缓存大文件之前调整文件系统预读设置。
- **临时文件。**如果您将临时文件放在某个文件系统上，而该系统启用了集成服务以通过大型文件读取数据并将数据写入临时文件，调整大型文件的文件系统读取和写入设置。
- **目标文件。**如果您将目标文件放在某个文件系统上，而该系统启用了集成服务以将大型文件写入到磁盘，针对同步大数据块写入调整文件系统。目标文件可以包括已分区会话的合并文件。由于网格上的已分区会话需要将文件写入磁盘，为优化锁定性能调整文件系统。

配置会话以分发文件

您可以手动配置会话来分发文件加载。当负载显著更改时或当您添加新会话或文件系统时，您可能需要编辑会话，包括将新节点添加到具有交叉装载共享文件系统的网格。

无需手动编辑会话，使用会话变量来将文件分发到不同的目录。这允许在必要时将会话文件重定向到不同的文件服务器。

使用以下准则使用会话变量：

- 反映业务逻辑的会话文件名和目录的名称变量。
- 在参数文件中，定义每个变量，以使文件加载均匀分布在所有可用的文件系统上。还可以定义特定于节点的变量。
- （可选）使用脚本实现重新配置自动化，以处理参数文件。

注意：当您使用一个脚本时，在参数文件中使用占位符，这样该脚本可以根据需要重新定义会话变量。

参数文件和脚本准则

当您创建参数文件和脚本时，请使用以下准则：

- 要轻松地保持会话文件位置的灵活性和控制性，请使用脚本来替换参数文件中的占位符。
- 定义文件位置时，考虑估计的文件大小和文件系统容量。
- 如果会话和工作流需要在同一时间访问与业务相关的文件，避免根据业务逻辑组织文件。例如，如果您在一个文件系统上存储加州的文件，在另一个文件系统上存储纽约的文件，如果会话需要同时访问这两种文件，可能出现瓶颈。
- 在可能的情况下，将相同的源、目标或查找的不同分区的文件放在不同文件系统上。

示例

在以下原始参数文件摘录中，占位符 “{fs}” 表示目录所在的文件系统，在使用前必须由一个脚本分配：

```
[SessionFFSrc_FFTgt_CA]
  $InputFile_driverInfo_CA={fs}/driverinfo_ca.dat
  $SubDir_processed_CA={fs}
# Session has Output file directory set to:
# $PmTargetFileDir/$SubDir_processed_CA
# This file is the input of SessionFFSrc_DBTgt_CA.
  $SubDir_RecordLkup_Cache_CA={fs}
# This session builds this persistent lookup cache to be used
  # by SessionFFSrc_DBTgt_CA.
# The Lookup cache directory name in the session is set to:
# $PmCacheDir/$SubDir_RecordLkup_Cache_CA
[SessionFFSrc_FFTgt_NY]
  $InputFile_driverInfo_NY={fs}/driverinfo_ny.dat
  $SubDir_processed_NY={fs}
[SessionFFSrc_DBTgt_CA]
  $SubDir_processed_CA={fs}
# session has Source file directory set to:
# $PmTargetFileDir/$SubDir_processed_CA
  $SubDir_RecordLkup_Cache_CA={fs}
# Use the persistent lookup cache built in SessionFFSrc_FFTgt_CA.
```

在以下参数文件摘录中，脚本使用适当的文件系统名称（如 file_system_1 和 file_system_2）替换了占位符：

```
[SessionFFSrc_FFTgt_CA]
  $InputFile_driverInfo_CA=file_system_1/driverinfo_ca.dat
  $SubDir_processed_CA=file_system_2
# Session has Output file directory set to:
# $PmTargetFileDir/$SubDir_processed_CA
# This file is the input of SessionFFSrc_DBTgt_CA.
  $SubDir_RecordLkup_Cache_CA=file_system_1
# This session builds this persistent lookup cache to be used
  # by SessionFFSrc_DBTgt_CA.
# The Lookup cache directory name in the session is set to:
# $PmCacheDir/$SubDir_RecordLkup_Cache_CA
[SessionFFSrc_FFTgt_NY]
  $InputFile_driverInfo_NY=file_system_2/driverinfo_ny.dat
  $SubDir_processed_NY=file_system_1
[SessionFFSrc_DBTgt_CA]
  $SubDir_processed_CA=file_system_1
# session has Source file directory set to:
# $PmTargetFileDir/$SubDir_processed_CA
  $SubDir_RecordLkup_Cache_CA=file_system_2
# Use the persistent lookup cache built in SessionFFSrc_FFTgt_CA.
```

优化序列生成器转换

要提高在带有序列生成器转换的网格上运行会话时的性能，请将缓存值的数目增加到每行数据一个数字。这样可减少主 DTM 进程和执行工作的 DTM 进程与存储库之间的通信量。主 DTM 和执行工作的 DTM 为每个缓存值进行一次通信。

例如，您有 150,000 行数据和 7 个序列生成器转换。缓存值的数量是 10。主 DTM 和执行工作的 DTM 通信 15,000 次。如果您将缓存值数量增加到 15,000，主 DTM 和执行工作的 DTM 通信 10 次。

第 9 章

优化 PowerCenter 组件

本章包括以下主题：

- [优化 PowerCenter 组件概览, 48](#)
- [优化 PowerCenter 存储库性能, 48](#)
- [优化集成服务性能, 50](#)

优化 PowerCenter 组件概览

您可以优化以下 PowerCenter 组件的性能：

- PowerCenter repository：PowerCenter 存储库
- 集成服务

如果您在多台计算机上运行 PowerCenter，请在不同计算机上运行存储库服务和集成服务。要加载大量数据，在更高处理计算机上运行集成服务。此外，在托管 PowerCenter 存储库的计算机上运行存储库服务。

优化 PowerCenter 存储库性能

完成以下任务以提高 PowerCenter 存储库的性能：

- 确保 PowerCenter 存储库与存储库服务进程位于同一台计算机上。
- 对对象查询中的条件进行排序。
- 如果您将 PowerCenter 存储库安装在一个 DB2 数据库上，为该存储库使用一个单节点表空间。
- 如果您将 PowerCenter 存储库安装在一个 DB2 或 Microsoft SQL Server 数据库上，优化该存储库的数据库架构。

存储库服务进程和存储库的位置

您可以优化配置为没有高可用性选项的存储库服务的性能。要优化性能，请确保存储库服务进程在存储库数据库所在的同一计算机上运行。

对对象查询中的条件进行排序

存储库服务处理含有多个条件的参数时，将按照您输入这些条件的顺序进行处理。为了达到预期效果并提高性能，请按照所希望的参数运行顺序输入参数。

使用单节点 DB2 数据库表空间

将 PowerCenter 存储库存储在单节点表空间中时，可以优化 IBM DB2 EEE 数据库中的存储库性能。当设置一个 IBM DB2 EEE 数据库时，数据库管理员可以在单个节点上定义数据库。

当表空间包含一个节点时，PowerCenter 客户端和集成服务访问存储库的速度比存储库表位于不同数据库节点的情况要快。

当创建、复制或还原存储库时，如果您不指定表空间名称，DB2 系统为每个存储库表指定默认表空间。DB2 系统可以指定也可以不指定单节点表空间。

优化数据库架构

如果您在管理控制台中为存储库服务启用了“优化数据库架构”选项，可以提高 IBM DB2 和 Microsoft SQL Server 数据库上的存储库性能。“优化数据库架构”选项使存储库服务尽可能在 Varchar(2000) 列中而不是在 CLOB 列中存储变化长度字符数据。使用 Varchar(2000) 列可以在以下方面提高存储库的性能：

- **减少磁盘访问。**PowerCenter 存储库直接在数据库表中的列内存储 Varchar 数据。它将 CLOB 数据作为引用存储到另一个表。要从存储库中检索 CLOB 数据，存储库服务必须访问一个数据库表以获取该引用，然后访问所引用的表来读取数据。要检索 Varchar 数据，存储库服务访问一个数据库表。
- **改进的缓存。**存储库数据库缓冲区管理器可以缓存 Varchar 列，但不缓存 CLOB 列。

优化数据库架构可以提高以下操作的存储库性能：

- 备份存储库
- 还原存储库
- 导出存储库对象
- 列出对象之间的相关性
- 部署文件夹

一般情况下，性能随存储库数据库和页面大小的增加相应提高。因此，优化数据库架构在更大的 PowerCenter 存储库中可以带来更大的性能改善。

您可以在创建存储库目录或备份及还原现有存储库时优化数据库架构。要优化数据库架构，存储库数据库必须满足以下页面大小要求：

- **IBM DB2。**数据库页面大小为 4 KB 或以上。至少存在一个页面大小为 16 KB 或以上的临时表空间。
- **Microsoft SQL Server。**数据库页面大小为 8 KB 或以上。

存储库服务的对象缓存

当您运行工作流时，存储库对象缓存可以提高性能。当存储库服务在内存中缓存对象时，集成服务可以更轻松地访问完成工作流运行所需的缓存对象。

要管理存储库服务的对象缓存，请配置以下属性：

属性	说明
存储库代理缓存	可选。启用存储库代理缓存，从而在集成服务重复运行多个会话时提高性能。如果您启用此属性，存储库服务进程缓存集成服务请求的元数据和说明存储库对象的元数据。默认值为 Yes。
代理缓存容量	可选。启用存储库代理缓存时缓存可以包含的对象数。如果在运行存储库服务进程的计算机上有可用的内存，可以增加对象的数目。默认值为 10,000。
允许使用代理缓存写入	可选。允许您使用 PowerCenter 客户端工具在启用存储库代理缓存时更改存储库中的元数据。当您启用写入操作时，存储库服务在每次通过 PowerCenter 客户端保存元数据时刷新缓存。如果您禁用写入，则无法通过 PowerCenter 客户端将元数据保存到存储库中，缓存不刷新。当您禁用写入时，集成服务仍然可以将会话和工作流元数据写入存储库。集成服务写入元数据时存储库服务不刷新缓存。默认值为 Yes。

优化弹性

配置应用程序服务的以下弹性相关属性，以提高存储库性能：

属性	说明
弹性超时限制	可选。该服务出于弹性目的而保持资源占用的最长时间。此属性对连接到服务的客户端设置了限制。超出限制的任何弹性超时都会在该限制时间截止。如果此属性的值为空，该值由域级别设置派生。默认值为空。
弹性超时	可选。该服务尝试建立或重新建立与另一服务的连接持续的时间段。如果此属性的值为空，该值由域级别设置派生。默认值为空。

优化集成服务性能

完成以下任务以提高集成服务性能：

- 为集成服务使用本地驱动程序而不是 ODBC 驱动程序。
- 如果字符数据是 7 位 ASCII 或 EBCDIC，在 ASCII 数据移动模式下运行集成服务。
- 为存储库服务缓存 PowerCenter 元数据。
- 以高可用性方式运行集成服务。

注意：当您以高可用性方式配置集成服务时，集成服务恢复可能由于暂时网络或计算机故障而失败的工作流和会话。要从工作流或会话恢复，集成服务将每个工作流和会话状态写入共享目录中的临时文件。这可能会降低性能。

使用本地和 ODBC 驱动程序

集成服务可以使用 ODBC 或本地驱动程序连接到数据库。使用本地驱动程序可提高性能。

以 ASCII 数据移动模式运行集成服务

当集成服务处理的所有字符数据都是 7 位 ASCII 或 EBCDIC 时，配置集成服务从而以 ASCII 数据移动模式运行。在 ASCII 模式中，集成服务使用一个字节来存储每个字符。当以 Unicode 模式运行集成服务时，它使用两个字节存储每个字符，可能会降低会话性能。

为存储库服务缓存 PowerCenter 元数据

您可以使用存储库代理缓存来提高 DTM 进程性能。如果启用存储库代理缓存，存储库服务缓存集成服务请求的元数据。当您缓存元数据时，集成服务为后续任务运行读取缓存，而不是从存储库中提取元数据。只缓存集成服务请求的元数据。

例如，运行带有 1,000 个会话的工作流。首次在启用缓存的情况下运行工作流，集成服务从存储库提取会话元数据。在后续运行工作流的过程中，存储库服务从缓存中提取会话元数据。这提高了 DTM 进程性能。

第 10 章

优化系统

本章包括以下主题：

- [优化系统概览, 52](#)
- [提高网络速度, 52](#)
- [使用多个 CPU, 53](#)
- [减少分页, 53](#)
- [使用处理器绑定, 53](#)

优化系统概览

因为会话依赖于效率低下的连接或过载的集成服务进程系统，性能通常会降低。路由器、交换机、网络协议以及由许多用户使用也可能会导致出现系统延迟。

源和目标数据库、源和目标文件系统以及域中的节点上磁盘访问速度缓慢可能会降低会话性能。请系统管理员评估计算机中的硬盘。

通过系统监视工具确定存在系统瓶颈之后，进行以下全局更改，可以提高所有会话的性能：

- **提高网络速度。**网络连接速度缓慢可能会降低会话性能。请系统管理员确定网络是否以最优速度运行。减少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 以并行运行多个会话，且并行运行多个管道分区。
- **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，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。

提高网络速度

集成服务的性能与网络连接相关。本地磁盘移动数据的速度比网络快 5 到 20 倍。考虑使用以下选项，以尽量减少网络活动，提高集成服务性能。

如果您在会话中使用平面文件作为源或目标，且集成服务在单个节点上运行，将文件存储在集成服务的同一台计算机上，以提高性能。如果您将平面文件存储在集成服务之外的计算机上，会话性能会依赖网络连接的性能。将文件移动到集成服务进程系统及增加磁盘空间可能会提高性能。

如果您使用关系源或目标数据库，请尽量减少源和目标数据库和集成服务进程之间的网络跃点数。将目标数据库移动到服务器系统上可能会提高集成服务性能。

如果运行包含多个分区的会话，请网络管理员分析网络，确保网络有足够带宽来处理从所有分区通过网络移动的数据。

使用多个 CPU

配置系统，以使用更多 CPU 来提高性能。多个 CPU 允许系统并行运行多个会话，并并行运行多个管道分区。

然而，额外的 CPU 可能会导致磁盘瓶颈。要防止磁盘瓶颈，尽量减少访问磁盘的进程数。访问磁盘的进程包括数据库功能和操作系统功能。并行会话或管道分区也要求磁盘访问。

减少分页

集成服务进程操作系统因执行特定操作而耗尽内存，且使用本地磁盘作为内存，则会出现分页。您可以释放更多内存，或增加物理内存来减少分页，并减少因分页导致的性能降低的情况。使用系统工具监视分页活动。

在以下情况下，您可能希望增加系统内存：

- 运行使用大型缓存查找的会话。
- 运行有许多分区的会话。

如果您无法释放内存，可能需要向系统添加内存。

使用处理器绑定

在多处理器 UNIX 环境中，如果您运行大量会话，集成服务可能会使用大量系统资源。其结果是，计算机上的其他应用程序可能没有足够的系统资源可用。您可以使用处理器绑定来根据集成服务进程节点控制处理器使用率。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。

在 Sun Solaris 环境中，系统管理员可以使用 `psrset` 命令创建和管理处理器集。然后，系统管理员可以使用 `pbind` 命令将集成服务绑定到一个处理器集，这样该处理器集仅运行该集成服务。Sun Solaris 环境还提供 `psrinfo` 命令来显示有关每个已配置处理器的详细信息，提供 `psradm` 命令来更改处理器的运行状态。有关详细信息，请参阅系统管理员与 Sun Solaris 文档。

在 AIX 环境中，系统管理员可以在 AIX 5L 中使用 Workload Manager，以在需求高峰时段管理系统资源。Workload Manager 可以分配资源和管理 CPU、内存和磁盘 I/O 带宽。有关详细信息，请参阅系统管理员与 AIX 文档。

第 11 章

使用管道分区

本章包括以下主题：

- [使用管道分区概览, 54](#)
- [为分区优化源数据库, 55](#)
- [为分区优化目标数据库, 57](#)

使用管道分区概览

在调整应用程序、数据库和系统以实现单分区性能最大化后，可能会发现该系统未充分利用。此时，您可以配置该会话，以包含两个或多个分区。

您可以使用管道分区来提高会话性能。增加分区或分区点数目可以增加线程的数量。因此，增加分区或分区点的数目也会增加集成服务的节点上的加载。如果集成服务的一个节点或多个节点包含充足的 CPU 带宽，在会话中并行处理多行数据可以增加会话性能。

注意：如果使用一个单节点集成服务，并且您在处理大量数据的会话中创建大量的分区或分区点，可能会使系统超负荷。

如果有分区选项，请执行以下任务来手动设置分区：

- 增加分区数量。
- 在管道中的特定点选择最佳分区类型。
- 使用多个 CPU。

增加分区数目

可以增加管道中的分区数量以提高会话性能。增加分区数将允许集成服务同时创建多个到源的连接并处理源数据的分区。

当会话使用文件源时，您可以配置它使用一个线程或多个线程来读取源。配置会话，以使用多个线程读取文件源，从而提高会话性能。集成服务创建到文件源的多个并发连接。

当创建一个会话时，Workflow Manager 验证分区映射中的每个管道。如果集成服务在处理已分区的数据时能保持数据的一致性，您可以在一个管道中指定多个分区。

向会话中添加分区时，请参考以下提示：

- **一次添加一个分区。**要获得最好的监视性能，请一次添加一个分区，并在添加每个分区之前，先记录会话设置。

- **设置 DTM 缓冲区内存。**当增加分区数量时，增加 DTM 缓冲区大小。如果该会话包含 n 个分区，将 DTM 缓冲区大小增大到带有一个分区的会话值的至少 n 倍。
- **设置序列生成器缓存值。**如果一个会话有 n 个分区，您不需要为序列生成器转换使用“缓存值数”属性。如果您将此值设置为大于 0 的值，请确保它至少为带有一个分区的会话原始值的至少 n 倍。
- **对源数据均匀分区。**配置每个分区以提取相同数量的行。
- **运行会话时监视系统。**如果 CPU 周期可用，您可以添加一个分区以提高性能。例如，如果系统有 20% 的空闲时间，您可能会有可用的 CPU 周期。
- **添加一个分区后监视系统。**如果 CPU 利用率没有上升，而 I/O 等待时间上升，或总数据转换率下降，那么有可能存在硬件或软件瓶颈。如果 I/O 等待时间上升非常显著，则检查系统的硬件瓶颈。否则，请检查数据库配置。

相关主题：

- [“缓冲区内存”页面上 38](#)

选择最好的分区类型

您可以在管道中的不同点指定不同分区类型，以提高会话性能。要优化会话性能，请为源和目标数据库使用数据库分区类型。您可以为 Oracle 和 IBM DB2 源和 IBM DB2 目标使用数据库分区。当使用源数据库分区时，集成服务查询数据库系统的表分区信息，并将数据提取到会话分区。当使用目标数据库分区时，集成服务将数据加载到相应的数据库分区节点。

您可以使用多个管道分区和数据库分区。要提高性能，请确保管道分区的数量等于数据库分区的数量。要提高子分区 Oracle 源的性能，请确保管道分区的数量等于数据库子分区的数量。

为提高性能，在管道中的以下分区点指定分区类型：

- **源限定符转换。**要同时从多个平面文件读取数据，请为源限定符转换中的每个平面文件指定一个分区。接受默认的分类型，传递。
- **筛选器转换。**由于源文件大小各不相同，每个分区处理不同的数据量。在筛选器转换设置分区点，并选择循环分区以平衡进入过滤器转换的加载。
- **排序器转换。**要消除排序器和汇总器转换中的重叠组，请在排序器转换使用哈希自动键分区。这会导致在排序器和汇总器转换处理行之前，集成服务将具有同一说明的所有项分组到同一分区。您可以在汇总器转换删除默认分区点。
- **目标。**由于目标表根据键范围分区，在目标指定键范围分区来优化将数据写入目标的性能。

使用多个 CPU

集成服务为一条管道并行执行读取、转换和写入处理。它可以在一个会话中处理管道的多个分区，并且可以并行处理多个会话。

如果您拥有对称多处理 (SMP) 平台，则可以使用多个 CPU 并发处理会话数据或数据的分区。这可带来性能的提升，因为实现了真正的并行。在单处理器平台上，这些任务共享 CPU，因此不存在并行。

集成服务可以使用多个 CPU 处理包含多个分区的会话。使用的 CPU 数量取决于以下因素，如分区数量、线程数量、可用 CPU 数量以及处理映射所需的数量或资源。

为分区优化源数据库

您可以添加分区以提高查询速度。通常情况下，读取器一侧的每个分区表示要处理的数据的一个子集。

完成以下任务来为分区优化源数据库。

- **调整数据库。**如果未正确调整数据库，创建分区可能无法加速会话。
- **Enable parallel queries.**某些数据库包含的一些选项必须设置为启用并行查询。请查看数据库文档了解这些选项。如果这些选项关闭，集成服务将以串行方式运行多个分区 SELECT 语句。
- **将数据分隔到不同表空间。**每个数据库都有一个选项可将数据分隔到不同表空间。如果数据库允许，使用 PowerCenter SQL 替代功能可提供从单个分区提取数据的查询。
- **对已排序数据进行分组。**可以对源数据进行分区和分组，以提高已排序联接器转换的性能。
- **最大化单排序查询。**

调整数据库

如果未正确调整数据库，结果可能无法加速会话。您可以测试数据库，确保已正确调整。

要验证数据库是否已正确调整，请执行以下操作：

1. 创建带有一个分区的管道。
2. 在工作流监视中测量读取器吞吐量。
3. 添加分区。
4. 验证吞吐量是否线性扩展。

例如，如果会话具有两个分区，读取器吞吐量速度应该是两倍。如果吞吐量不能线性扩展，您可能需要调整数据库。

对已排序数据进行分组

还可以对源数据进行分区和分组，以提高已排序联接器转换的性能。将分区点放在排序转换之前，以维护分组，并在每个组内对数据进行排序。

要对数据进行分组，确保将具有相同键值的行路由到同一分区。确保在分区之间均匀进行数据分组和分布的最好方法是在排序源之前添加一个哈希自动键或键范围分区点。

优化单排序查询

要优化数据库的单排序查询，请考虑以下启用并行化的调整选项：

- **检查执行自动调整的配置参数。**例如，Oracle 有一个称为 parallel_automatic_tuning 的参数。
- **请确保启用了内并行性。**内并行性是在单个查询上运行多个线程的能力。例如，在 Oracle 中，查看 parallel_adaptive_multi_user。在 DB2 中，查看 intra_parallel。
- **验证可用于并行执行的并行进程的最大数目。**例如，在 Oracle 中，查看 parallel_max_servers。在 DB2 中，查看 max_agents。
- **验证并行化中使用的各种资源的大小。**例如，Oracle 有 large_pool_size、shared_pool_size、hash_area_size、parallel_execution_message_size 和 optimizer_percent_parallel 等参数。DB2 有 dft_fetch_size、fcm_num_buffers、sort_heap 等配置参数。
- **验证并行度。**您可能可以通过在表或查询上使用一个数据库配置参数或选项来设置此选项。例如，Oracle 具有参数 parallel_threads_per_cpu 和 optimizer_percent_parallel。DB2 具有 dft_prefetch_size、dft_degree 和 max_query_degree 等配置参数。
- **关闭可能会影响数据库可扩展性的选项。**例如，禁用 Oracle 的存档日志记录 and 定时统计信息。

有关数据库调整选项的完整列表，请参阅数据库文档。

为分区优化目标数据库

如果一个会话包含多个分区，每个分区的吞吐量应该与单个分区会话的吞吐量相同。如果看不到这种关联性，那么数据库可能将行以串行方式插入到数据库。

要确保数据库并行插入行，请检查目标数据库中的以下配置选项：

- **设置数据库中的选项以启用并行插入。**例如，设置 `db_writer_processes`，DB2 在 Oracle 数据库中具有 `max_agents` 选项以启用并行插入。有些数据库可能在默认情况下启用这些选项。
- **考虑对目标表进行分区。**如果可能，尝试将每个分区写入使用路由器转换进行此操作的单个数据库分区。将数据库分区放在不同磁盘，以防止管道分区之间的 I/O 争用。
- **在数据库中设置选项以增强数据库可扩展性。**例如，在 Oracle 数据库中禁用存档日志记录和定时统计信息，以增强可扩展性。

附录 A

性能计数器

本附录包括以下主题：

- [性能计数器概览, 58](#)
- [Errorrows 计数器, 58](#)
- [Readfromcache 和 Writetocache 计数器, 58](#)
- [Readfromdisk 和 Writetodisk 计数器, 59](#)
- [Rowsinlookupcache 计数器, 59](#)

性能计数器概览

所有转换都有计数器。集成服务跟踪每个转换的输入行、输出行和错误行的数目。一些转换具有性能计数器。可以使用以下性能计数器提高会话性能：

- Errorrows
- Readfromcache 和 Writetocache
- Readfromdisk 和 Writetodisk
- Rowsinlookupcache

Errorrows 计数器

转换错误影响会话性能。如果转换在任何 *Transformation_errorrows* 计数器中有大量的错误行，您可以消除这些错误，从而提高性能。

相关主题：

- [“消除转换错误” 页面上 36](#)

Readfromcache 和 Writetocache 计数器

如果会话包含汇总器、等级或联接器转换，请检查 *Transformation_readfromcache* 和 *Transformation_writetocache* 计数器以及 *Transformation_readfromdisk* 和 *Transformation_writetodisk*

计数器来分析集成服务如何从磁盘读取或向磁盘写入。要在会话运行时查看会话性能详细信息，请在工作流监视中右键单击该会话并选择“属性”。单击详细信息对话框中的“属性”选项卡。

要分析磁盘访问，首先计算命中率或错失率。命中率表示集成服务在缓存中执行的读取或写入操作数。

错失率表示集成服务在磁盘执行的读取或写入操作数。

使用以下公式来计算缓存错失率：

$$\frac{[(\# \text{ of reads from disk}) + (\# \text{ of writes to disk})]}{[(\# \text{ of reads from memory cache}) + (\# \text{ of writes to memory cache})]}$$

使用以下公式来计算缓存命中率：

$$[1 - \text{Cache Miss ratio}]$$

要尽量减少对磁盘的读取和写入操作，请增加缓存大小。优化缓存命中率是 1。

Readfromdisk 和 Writetodisk 计数器

如果会话包含汇总器、等级或联接器转换，请检查每个 *Transformation_readfromdisk* 和 *Transformation_writetodisk* 计数器。要在会话运行时查看会话性能详细信息，请在工作流监视中右键单击该会话并选择“属性”。单击详细信息对话框中的“属性”选项卡。

如果这些计数器显示任何非零数字，可以增加缓存大小以提高会话性能。集成服务使用索引缓存来存储组信息，使用数据缓存存储转换的数据，这些数据一般较大。因此，尽管索引缓存和数据缓存大小都会影响性能，您可能仍然需要相比索引缓存大小更大幅度地增加数据缓存大小。然而，如果处理的数据量大于可用内存，可以增加索引缓存大小以提高性能。

例如，集成服务使用 100 MB 存储索引缓存，使用 500 MB 存储数据缓存。如果在每个索引和数据缓存随机分布 200 次访问，您可以通过以下方式配置缓存：

- 要优化性能，为索引缓存分配 100 MB，为数据缓存分配 200 MB。集成服务通过索引缓存访问 100% 的数据，通过数据缓存访问 40% 的数据。集成服务始终访问索引缓存，而有 120 次不访问数据缓存。因此，获得访问的数据百分比是 70%。
- 为索引缓存分配 50 MB，为数据缓存分配 250 MB。集成服务访问索引缓存中 50% 的数据以及数据缓存中 50% 的数据。集成服务有 100 次不访问索引和数据缓存。因此，访问的数据百分比为 50%。

如果会话执行增量汇总，集成服务在会话期间从本地磁盘读取历史汇总数据，在保存历史数据时写入磁盘。其结果是，*Aggregator_readtodisk* 和 *Aggregator_writetodisk* 计数器显示不为零的数字。

然而，由于集成服务在会话结束时将历史数据写入一个文件，您仍可以在会话期间评估计数器。如果计数器在运行会话期间显示非零数字，您可以调整缓存大小以提高性能。然而，分配或取消分配内存存在相关成本，因此如果您知道集成服务将处理的数据量，不增加缓存大小，以容纳更多的数据量。

Rowsinlookupcache 计数器

多个查找可以降低会话的性能。要提高会话性能，可以调整大型查找表的查询表达式。

相关主题：

- [“优化多个查找”页面上 33](#)

索引

A

ASCII 模式
性能 [51](#)

B

绑定
 处理器 [53](#)
表达式
 计算 [28](#)
 调整 [26](#)
 替换为本地变量 [27](#)
表空间
 用于 DB2 的最佳类型 [49](#)

C

参数文件
 性能准则 [46](#)
 最佳存储 [44](#)
操作
 数值与字符串 [27](#)
查询
 调整关系源 [22](#)
查找 SQL 替代选项
 减少缓存大小 [32](#)
查找条件
 匹配 [32](#)
 优化 [33](#)
查找转换
 调整 [31](#)
 通过 ORDER BY 语句优化 [32](#)
 通过并发缓存优化 [32](#)
 通过高内存计算机优化 [33](#)
 通过缓存优化 [31](#)
 通过减少缓存进行优化 [32](#)
 通过数据库驱动程序优化 [31](#)
 通过索引优化 [33](#)
 优化 [59](#)
 优化查找条件 [33](#)
 优化查找条件匹配 [32](#)
 优化多个查找表达式 [33](#)
持久性缓存
 用于查找 [32](#)
持久性缓存文件
 配置准则 [45](#)
 最佳存储 [44](#)
传递映射
 调整 [25](#)
处理器
 绑定 [53](#)
磁盘
 访问, 最小化 [53](#)

CPU

 多个并发工作流 [53](#)
 多个管道分区 [55](#)
存储库
 数据库架构, 优化 [49](#)
存储库服务
 缓存存储库元数据 [51](#)
存储库服务进程
 最佳位置 [48](#)
错误
 最小化跟踪级别 [41](#)
错误跟踪
 请参阅跟踪级别[错误跟踪
 aaa] [41](#)

D

带分隔符的平面文件
 源 [25](#)
DB2
 PowerCenter 存储库性能 [49](#)
DECODE 函数
 用于优化 [27](#)
 与 Lookup 函数比较 [27](#)
等级转换
 性能详细信息 [58](#), [59](#)
DTM 缓冲区
 最佳池大小 [38](#)
端口
 连接, 限制 [39](#)
对象查询
 对条件排序 [48](#)
读取测试映射
 源瓶颈, 标识 [15](#)

F

方法
 筛选数据 [25](#)
繁忙时间
 线程统计信息 [13](#)
FastExport
 用于 Teradata 源 [23](#)
非持久性缓存
 最佳文件存储 [44](#)
分区
 添加 [54](#)
分区类型
 优化 [55](#)
分页
 减少 [53](#)
分组依据端口
 优化汇总器转换 [29](#)

G

- 高可用性
 - 群集文件系统 [44](#)
- 跟踪级别
 - 最小化 [41](#)
- 共享缓存
 - 用于查找 [32](#)
- 共享文件系统
 - CPU, 平衡 [45](#)
 - 低带宽 [44](#)
 - 服务器加载, 分布 [45](#)
 - 概览 [44](#)
 - 高带宽 [44](#)
 - 配置 [44](#)
- 工作流
 - 并发 [38](#)
- 工作流日志文件
 - 禁用 [41](#)
- 管道
 - 数据流监视 [58](#)
- 管道分区
 - 多个 CPU [55](#)
 - 添加分区 [54](#)
 - 调整源数据库 [56](#)
 - 优化目标数据库 [57](#)
 - 优化性能 [54](#)
 - 优化源数据库 [55](#)
 - 最佳分区类型 [55](#)
- 规范器转换
 - 调整 [34](#)

H

- 函数
 - DECODE 与 LOOKUP [27](#)
 - 与运算符比较 [27](#)
- 函数调用
 - 最小化自定义转换 [30](#)
- 缓冲
 - 数据 [28](#)
- 缓冲区长度
 - 优化设置 [24](#)
- 缓冲区块大小
 - 优化 [39](#)
- 缓冲区内内存
 - 分配 [38](#)
- 缓存
 - 存储库元数据 [51](#)
 - 减少缓存的行 [32](#)
 - 调整 [39](#)
 - 序列值 [34](#)
 - 最佳大小 [40](#)
 - 最佳位置 [40](#)
- 缓存目录
 - 共享 [40](#)
- 缓存文件
 - 最佳存储 [44](#)
- 会话
 - 并发 [38](#)
 - 瓶颈, 标识 [16](#)
 - 瓶颈, 消除 [16](#)
 - 瓶颈, 原因 [16](#)
 - 调整 [37](#)
 - 网格 [37](#)
 - 下推优化 [38](#)

- 会话后电子邮件
 - 性能 [42](#)
- 会话日志文件
 - 禁用 [41](#)
- 汇总函数
 - 最小化调用次数 [26](#)
- 汇总器转换
 - 调整 [29](#)
 - 通过分组依据端口优化 [29](#)
 - 通过筛选器优化 [30](#)
 - 通过已排序输入优化 [30](#)
 - 通过有限的端口连接优化 [30](#)
 - 性能详细信息 [58](#), [59](#)
 - 增量汇总 [30](#)

I

- IBM DB2
 - 存储库数据库架构, 优化 [49](#)
- IIF 表达式
 - 调整 [27](#)
- IPC 协议
 - Oracle 源 [23](#)

J

- 检查点时间间隔
 - 增加 [20](#)
- 键约束
 - 删除 [19](#)
- 集成服务
 - 调整 [50](#)
 - 提交时间间隔 [40](#)
 - 网格 [37](#)
 - 最佳数据库驱动程序 [50](#)
- 计算
 - 表达式 [28](#)
- 局部变量
 - 替换表达式 [27](#)

K

- 空格
 - 尾随, 删除 [27](#)
- 空闲时间
 - 线程统计信息 [13](#)

L

- 联接
 - 在数据库中 [30](#)
- 联接器转换
 - 调整 [30](#)
 - 性能详细信息 [58](#), [59](#)
 - 已排序数据 [30](#)
 - 指定主源 [30](#)
- 连续合并
 - 最佳文件存储 [44](#)
- 临时文件
 - 最佳存储 [44](#)
- LOOKUP 函数
 - 与 DECODE 函数比较 [27](#)
 - 最小化优化 [27](#)

M

Microsoft SQL Server
存储库数据库架构, 优化 [49](#)
内存中的数据库 [23](#)
目标
分配缓冲区内存 [38](#)
瓶颈, 消除 [14](#)
瓶颈, 原因 [14](#)
确定瓶颈 [14](#)
目标文件
最佳存储 [44](#)

N

内存
64 位 PowerCenter [40](#)
缓冲区 [38](#)
Microsoft SQL Server 数据库 [23](#)
Sybase ASE 数据库 [23](#)
增加 [53](#)

O

Oracle
IPC 协议 [23](#)
调整目标 [21](#)
外部加载器 [20](#)
优化连接 [23](#)
ORDER BY
优化查找转换 [32](#)

P

排序器转换
调整 [34](#)
通过内存分配优化 [34](#)
优化分区目录 [35](#)
批量加载
调整关系目标 [20](#)
瓶颈
标识 [12](#)
会话 [16](#)
目标 [14](#)
线程统计信息 [13](#)
消除 [12](#)
系统 [16](#)
映射 [15](#)
源 [14](#)
在 UNIX 中 [17](#)
在 Windows 中 [17](#)
平面文件
带分隔符的源文件 [25](#)
缓冲区长度 [24](#)
优化源 [24](#)
与 XML 文件比较 [25](#)
最佳存储位置 [52](#)
平面文件日志记录
会话后电子邮件 [42](#)
PowerCenter 存储库
DB2 的性能 [49](#)
调整 [48](#)
最佳位置 [48](#)

Q

群集文件系统
另请参阅共享文件系统[群集文件系统
aaa] [44](#)
高可用性 [44](#)

R

日志文件
最佳存储 [44](#)

S

筛选
数据 [25](#)
源数据 [35](#)
筛选器
源 [22](#)
筛选器转换
源瓶颈, 标识 [15](#)
删除
索引和键约束 [19](#)
尾随空格 [27](#)
实时会话
性能, 提高 [41](#)
刷新延迟
性能, 提高 [41](#)
数据缓存
连接的端口 [39](#)
最佳大小 [40](#)
最佳位置 [40](#)
数据库
检查点时间间隔 [20](#)
联接 [30](#)
调整 Oracle 目标 [21](#)
调整单排序查询 [56](#)
调整源 [22](#)
网络包大小 [21](#), [23](#)
优化目标以进行分区 [57](#)
优化源以进行分区 [55](#)
最大程度减少死锁 [20](#)
数据库查询
源瓶颈, 标识 [15](#)
数据库驱动程序
针对集成服务最佳 [50](#)
数据类型
变长字符型 [27](#)
优化转换 [26](#)
字符 [27](#)
数据流
监视 [58](#)
优化 [58](#)
数据移动模式
优化 [51](#)
数值操作
与字符串操作比较 [27](#)
死锁
最小化 [20](#)
SQL 转换
调整 [35](#)
索引
删除 [19](#)
用于查找表 [33](#)
索引缓存
最佳大小 [40](#)

索引缓存 (续)
最佳位置 [40](#)
Sybase ASE
内存中的数据库 [23](#)
Sybase IQ
外部加载器 [20](#)

T

Teradata FastExport
源性能 [23](#)
调整
表达式 [26](#)
查找转换 [31](#)
关系源 [22](#)
规范器转换 [34](#)
缓存 [39](#)
会话 [37](#)
汇总器转换 [29](#)
集成服务 [50](#)
联接器转换 [30](#)
排序器转换 [34](#)
PowerCenter 存储库 [48](#)
SQL 转换 [35](#)
网络 [52](#)
系统 [52](#)
XML 转换 [35](#)
序列生成器转换 [34](#)
映射 [24](#)
源限定符转换 [35](#)
转换 [29](#)
自定义转换 [30](#)
提交时间间隔
会话性能 [40](#)
提取
映射的通用逻辑 [26](#)

U

UNIX
处理器绑定 [53](#)
瓶颈, 消除 [17](#)
系统瓶颈 [17](#)

V

Varchar 数据类型
删除尾随空白 [27](#)

W

外部过程转换
阻止数据 [28](#)
外部加载器
性能 [20](#)
网络
节点瓶颈 [45](#)
性能 [37](#), [43](#)
序列生成器性能, 提高 [47](#)
最佳存储位置 [43](#)
网络上的会话
序列生成器性能, 提高 [47](#)
网络
调整 [52](#)

网络 (续)
提高速度 [52](#)
网络包
增加 [21](#), [23](#)
网络文件系统
请参阅共享文件系统[网络文件系统
aaa] [44](#)
文件存储
本地 [43](#)
共享文件系统 [43](#)
类型 [43](#)
文件共享
群集文件系统 [44](#)
网络文件系统 [44](#)
文件系统
共享, 配置 [44](#)
群集 [44](#)
网络 [44](#)
Windows
瓶颈 [17](#)
瓶颈, 消除 [17](#)

X

线程
繁忙时间 [13](#)
空闲时间 [13](#)
瓶颈, 标识 [13](#)
线程工作时间 [13](#)
运行时间 [13](#)
线程统计信息
瓶颈, 标识 [13](#)
瓶颈, 消除 [13](#)
下推优化
性能 [38](#)
性能
存储库数据库架构, 优化 [49](#)
实时会话 [41](#)
刷新延迟 [41](#)
调整, 概览 [11](#)
性能计数器
类型 [58](#)
Rowsinlookupcache [59](#)
Transformation_errorrows [58](#)
Transformation_readfromcache [58](#)
Transformation_readfromdisk [59](#)
Transformation_writetocache [58](#)
Transformation_writetodisk [59](#)
系统
瓶颈, 使用工作流监视标识 [17](#)
瓶颈, 消除 [17](#)
瓶颈, 原因 [16](#)
调整 [52](#)
UNIX 中的瓶颈, 标识 [17](#)
Windows 中的瓶颈, 标识 [17](#)
XML 文件
与平面文件比较 [25](#)
XML 源
分配缓冲区内内存 [38](#)
XML 转换
调整 [35](#)
选择相异
筛选源数据 [35](#)
序列生成器转换
可重用 [34](#)
调整 [34](#)
网络性能 [47](#)

Y

- 页面大小
 - 优化存储库架构的最低要求 [49](#)
- 一次性读取
 - 定义 [25](#)
- 映射
 - 传递映射, 调整 [25](#)
 - 瓶颈, 标识 [16](#)
 - 瓶颈, 消除 [16](#)
 - 调整 [24](#)
 - 提取通用逻辑 [26](#)
 - 一次性读取 [25](#)
- 已排序输入
 - 优化汇总器转换 [30](#)
- 源
 - 关系, 调整 [22](#)
 - 瓶颈, 消除 [15](#)
 - 瓶颈, 原因 [14](#)
 - 确定瓶颈 [14](#)
 - 筛选器 [22](#)
 - 调整查询 [22](#)
- 源文件
 - 平面与 XML [25](#)
 - 最佳存储 [44](#)
- 源限定符转换
 - 调整 [35](#)
- 运算符
 - 与函数比较 [27](#)
- 运行时间
 - 线程统计信息 [13](#)

Z

- 暂存区域
 - 删除 [41](#)
- 增量汇总
 - 优化汇总器转换 [30](#)
- 转换
 - 数据类型 [26](#)
 - 调整 [29](#)
 - 消除错误 [36](#)
 - 优化 [58](#)
- 转换线程
 - 线程工作时间 [13](#)
- 自定义转换
 - 处理数据块 [30](#)
 - 调整 [30](#)
 - 最小化函数调用 [30](#)
- 字符串操作
 - 与数值操作比较 [27](#)
 - 最小化 [27](#)
- 字符数据类型
 - 删除尾随空白 [27](#)
- 最佳文件存储
 - 参数文件 [44](#)
 - 非持久性缓存文件 [44](#)
 - 临时文件 [44](#)
 - 目标文件 [44](#)
 - 日志文件 [44](#)
 - 源文件 [44](#)
- 最小化
 - 汇总函数调用 [26](#)