

Data Integrator 性能测试报告

执行摘要

本测试白皮书来自 Business Objects, an SAP company, 本白皮书证明 Business Objects Data Integrator 和 Data Services 软件能够实现非常高的运行性能。尽管这个测试是用 Data Integrator 来进行的, 但其结果对 Data Services XI 3.0 也同样适用, 这是因为 Data Services 就是建立在 Data Integrator XI 的性能基础之上的, 它是第一个将数据整合和数据质量功能整合在一起的应用软件。

目前大部分抽取、转换和加载软件 (ETL 软件) 的基础测试并不反映数据在生产环境下是如何整合和处理的。例如, 这种测试通常把注意力放在对大数据量的原始数据从一种文件格式迁移到一个数据库中。如果只是想看简单的数据迁移任务的性能, 这种做法当然没问题, 但企业中数据库和应用软件中的数据可不像大数据文件那么简单。

Business Objects 相信演示 ETL 性能的最佳方法就是用大多数企业组织内常用的场景来处理大数据量。我们创建并进行的这个 ETL 基准测试来证明 Business Objects Data Integrator 能够扩展至最大量的企业数据量, 并且使用真实的场景。

我们的基准测试从一个标准的订单数据库中抽取大于 1TB 的数据量, 并将其装入一个数据仓库系统的星型模型中。这个基准测试包括针对不同任务的一组测试, 这组测试符合将数据装入数据仓库过程中的典型人物, 包括对客户数据的历史数据保存, 以及执行数据校验规则。

测试结果显示, Data Integrator 每小时能够处理超过 2TB 的数据。这就是企业级性能, 别忘了在这个测试中包含了对建立一个星型结果的标准逻辑处理。

在一个独立的测试中，DI 向一个事实表中装载 45 亿条记录只用了 43 分钟，吞吐率超过每秒钟 170 万条。这些结果表明，DI 能够满足绝大部分大型企业级的 ETL 作业任务。但这还仅仅是两个重要的要素之一。

这个测试数据没有表达出来的另外一个要素是，使用 DI 实现这个级别的性能是多么容易。使用其他大多数 ETL 产品，开发人员必须花费数小时来仔细构建并行 JOB，管理内存并且调试 ETL 执行引擎。而使用 DI 创建这个基准测试作业所用的时间总共花费了 14 分钟，而且并没有进行什么特别的调优或使用什么特别的特性——仅仅使用了标准的选项来将并行处理的进程数量进行调整。

DI 的另外一个独特的性能特性是将数据放在 ETL 引擎中执行或者将转换任务推送给数据库平台的能力。一个智能优化器自动决定哪种办法能够实现最佳性能。由于使用了并行执行和智能推送的优化机制的组合，使得在 DI 中创建一个大数据量的作业非常简单，而不像其他大部分 ETL 工具中那么复杂。

在评估 ETL 工具时，将实现高性能 ETL Job 所花费的时间，以及所需的知识水平纳入考量范围非常重要。一个易用使用的产品能让工作做起来更快，让你的开发人员工作更高效，从而加速你的项目进度。开发这些高性能的 Job 所花费的时间，因该作为您评估 ETL 性能的一个重要因素。

企业 ETL 需求

在检视这个基准测试的细节之前，我们先看一下企业 ETL 的处理需求，以及一个真实的基准测试由什么构成。

数据仓库的 ETL 工作量

ETL 并不像大部分基准测试描述的那样简单，对于一个数据仓库来说，ETL 实际上有两种工作量需要进行测试。其中之一是将历史数据加载到一个空的数据仓库中去，这个过程称为初始化加载过程。

初始化加载的典型特征是数据量非常庞大，通常是几年的数据需要加载。这些数据需要加载到一个空的数据仓库中去，并且假定大部分的数据校验工作在加载之前都

已经完成了。如果你的 ETL 工具不能有效处理大数据量的话，你可能需要花费数个星期来加载数据，然后的数据仓库才能启动运行。

有时候由于数据需求的扩展，或者需要加载额外的历史数据，而需要向现有的数据仓库中重新加载数据，这种情况并不少见。还有一种情况是，需要将数据仓库进行合并，比如将两个或者多个数据集市或数据仓库合并成一个数据库，这种情形也会产生重新加载的需求。重新加载和合并与初始化加载的需求其实是一样的。

另外一种工作任务是增量加载，增量加载处理自从上次加载之后变化的数据。增量加载的特点是数据量比较小，但通常会要求在一个更小的时间窗内处理完毕，这意味着几十数据量变小了，而对 ETL 的性能要求却通常提高了。

增量数据加载通常会面临数据量波动的挑战，这种情况可被预测，比如由于季节性的数据增加，或者由于不可预见的事件造成的数据量尖峰。系统必须能够在可预测的方式运行，以便即使出现数据尖峰，数据仍然能够在时间窗内加载进来。

有时候还能预见流过系统的数据量会随着时间的推移而增加，行业平均值是每年增加 20%，当你计划 ETL 扩展性的时候，这种数据量的逐年增加必须予以考虑，无论是平台角度，还是从 ETL 工作流的设计角度均需考虑。

测量实际性能

获得实际的性能指标是个挑战，在没有标准的基准测试标准的情况下，你如何能够评估 ETL 的性能呢？甚至连测试指标的选择也成为一个问题。你可以查看数据处理的行数，但如果一个厂商采用了毫无意义的 ETL 工作流，而另外一家则并不把数据源假想成完美无暇而采用了更加符合实际情况的工作流呢，只观察数据行数还有意义吗？

这并不是说基准测试毫无用处，只要测试用例有实际意义，基准测试还是能够评判一个产品是否能具备合理的性能的。

当你查看一个 ETL 基准测试的时候，你可以通过检查如下关键要素来确保这个测试能够反映真实情况的性能。

真实的 ETL 场景

执行的测试用例应该和在数据仓库构建过程中的任务一致，这种任务包括对数据执行规则校验并保存异常数据、从其他表中进行字典查找（lookup）、而维表执行标准函数操作。标准函数则包括生成代理键、管理维表中的变化历史、执行汇总，以及在事实表中映射键值。

源和目标

许多 ETL 基准测试使用预先抽取出来的平面文件作为数据源，而在实际情况中，从数据库中直接抽取数据比从文件中取数更加常见。使用数据库进行抽取时，还通常要求 ETL 作业对数据进行联合和分组，而这些操作对文件来说通常并不存在。你其实需要的是一个端到端的测试，将数据抽取出来，再送到数据库中去，否则你就只是在测试把文件加载到数据库中有多快，好比是一个与 ETL 毫不相干的测试。

数据量和复杂程度

一个 ETL 基准测试需要使用多种方式来测试数据量的处理能力。大数据量测试意味着不仅仅是将上 TB 的数据从单一的表中进行迁移出来。当你需要把几个大表进行关联，或者一个大表中的数据值需要从小表中进行查找（lookup）转换，复杂性问题就显现出来了。只有你认识到有的表很宽，因为它有很多列，而有些表有较少的列而相对较窄，基于数据行数的数据量测试才是一个好的测试。

工作量

基准测试应该重现初始化加载和增量加载两种情形下的工作任务。如果基准测试只是向一个空的数据库中加载了历史数据，那么这个测试只验证了部分的性能问题。

企业ETL性能特性

评估一个基准测试最难的事情是让测试结果对你来说变得有意义，将其结果对应到你自己的环境中。虽然根据测试结果推断在一个较小的硬件平台上的运行结果并不难，但要想重复厂商所作的特别的优化却几乎是不可能的，这就使得推断结果并不可靠。

要想对测试结果的真实性有信心，ETL 工具需要具备随时可用的企业级特性，来确保你能够实现相同的性能水平，这些特性对于现实中的高性能非常重要。

并行处理

并行处理有很多种形式。最基本的一个级别，工具应该允许开发人员来启用并行执行机制，使得一个作业能够充分使用所有的处理器（CPU）。更高级的并行特性包括调整一个作业中的工作流级别以及单个转换器级别上调整并行度。这种调整的结果是，如果一个 ETL 过程存在性能瓶颈，一个简单的调整就能解决问题。

下推处理

下推处理的意思是智能地将一些处理过程从 ETL 引擎推送给数据库平台引擎来执行。除了将数据在不同服务器之间的传送将降至最低之外，下推处理还能够将一些特定的转换工作放到最有效率的地方去执行，从而降低 ETL 环境的总体成本。例如，通常将汇总数据的操作交给数据库完成比交给 ETL 引擎完成更有效率。

下推操作允许 ETL 服务器执行最有效的处理，无论是使用自己的引擎还是使用数据库引擎。ETL 工具提供一些选项，由你决定将下推操作提交给数据源，或者数据目标，或者由 ETL 引擎完成，这种选择为获得你所需的性能提供了更灵活的手段。

网格计算

在过去的几年中，实现扩展性的常规思路发生了变化，从使用一台超大的服务器，转而使用更多数量的更小、更便宜的服务器来实现扩展性。一个企业级 ETL 工具要求他能够在多台服务器上运行，如同在单台服务器上运行一样容易。它还应该能够提供集中的管理功能来管理分布式环境，将作业自动分配给负载最轻的服务器，并且对每个大型的处理需求使用计算网格。

易用性

易用性对于在真实环境中获得良好性能至关重要。高级的并行处理或者下推特性只有在常被使用的前提下才能发挥作用，如果这些特性非常难用，大部分开发人员就会避免使用。

一个具备高级特性的 ETL 产品需要智能优化功能，以便由它自己决定是将操作推送给数据库还是并行执行。这个过程需要尽可能做到透明，你可不需要一个只有具有博士学位的人才能使用的性能特性。

请将上述这些要点记在心中，下面我们看一下基准测试。

基准测试概要

Business Objects 根据客户的数据中心典型的数据仓库抽取和加载过程创建了这个企业 ETL 基准测试。这个基准测试由一系列常规的任务组成，使用了产品的不同组件。这些任务用来表明在一个数据仓库项目中面临的超大量数据的处理性能。

场景

这个业务场景使用来自一个传统的交易数据库模型的订单数据库来构建一个数据库，数据源的数据量为 1TB。数据仓库采用星型模式，用来存储每个客户的每笔交易订单。在这个场景中，整个基准测试有三个不同的任务：

1. 第一个测试构建一个没有历史数据的产品维。我们的处理任务是将产品信息与库存记录连接起来判断其总的库存量和每个产品的可用数量。

第二个测试是构建一个客户维，这个过程保留已有客户记录的变化情况，比如地址变化。这个作业执行一个校验规则以检查一个名称是否已经存在，并将校验失败的记录放到一个异常数据区域。另外一个规则检查信用等级，检查不良等级的记录单允许这些数据进一步处理。

第三个测试执行一个订单细节事实表的初始化加载。所有的客户订单记录在一个批作业中进行处理来填充事实表。

不像许多 ETL 基准测试，这些测试评估了更加真实的数据仓库处理过程，比如在一个维表中保留历史，对数据源执行校验并保存异常数据，以及将范式化的数据映射生成一个维模型（星型模式）。

这个基准测试并不是批量加载文本文件，这一点非常重要。这个测试使用了常见的真实场景，将数据从一个服务器的交易数据库中抽取出来，并将其加载到网络上的另外一台服务器上的数据仓库中。

对结果的分析

让我们看一下上述测试的细节情况，以便更好地理解这些测试反映了 BO Data Integrator 性能的哪些方面。

测试1：加载产品维

在这个测试中，先将目标产品维表中的数据全部删除，然后两个源表进行关联，产生了15亿行数据。这些行进行汇总，产生10万条产品数据，这些数据再插入到目标表中。期间执行了一些运算来生成全部库存和可用库存数，这些结果也插入到目标产品维表中。

这行这个测试过程处理了 404GB 数据源，仅仅耗时 12 分钟，相当于每小时吞吐量 2TB，这可是个创纪录的数据。请看表 1。

Table 1. Test 1 – Load the Product Dimension	
Elapsed Time	12 minutes
Number of Rows Processed	1.5 billion
Number of Rows Inserted at Target	100,000
Gigabytes Processed	404 GB
Gigabytes Per Hour	2020 GB / hr
Rows Per Second	1,590,668
Time to Develop the ETL Job	3 minutes

测试 2：加载客户维

客户维有 4.5 亿条记录，大约 269GB，为了保留目标表中已有数据的变化，必须为目标表中现存的每个客户都插入一条新的记录，在新记录上还需做出标识来表明这是现有客户并将原来的记录标识接触，而 Valid-from 和 Valid-to 日期必须写入原记录和新记录。这个过程包括对每个进来的数据进行数据规则校验，产生客户键值，并从其他表中查找数据（Lookup）。

这个测试比上第一个测试处理的数据量小，但没有进行汇总，测试必须向目标表中插入大量数据。这个作业的逻辑更加复杂，使用了查找和数据校验操作。即便增加了复杂性，这个测试在 32 分钟内处理了 269GB 数据，见表 2：

Table 2. Test 2 – Load the Customer Dimension	
Elapsed Time	32 minutes
Number of Rows Processed	450 million
Number of Rows Inserted at Target	450 million
Gigabytes Processed	269 GB
Gigabytes Per Hour	522 GB / hr
Rows Per Second	231,600
Time to Develop the ETL Job	5 minutes

测试 3： 加载交易明细事实表

这个测试演示了向事实表中加载超大量的数据，45 亿条记录。这个测试包含了所有构建一个事实表所需的复杂操作，比如大表的关联，多个事实表的键值查找，执行计算，设定默认值等。

这个测试比前一次测试数据量大 10 倍，向一个事实表中加载了 340GB 数据。处理这 45 亿条记录花了 43 分钟，相当于每秒钟 170 万条记录。尽管数据行数增加了 10 倍，数据量增加了 20%，耗时却只比上一个测试多了 10%，还不算更加复杂的处理逻辑。见表 3。

Table 3. Test 3 – Load the Fact Table	
Elapsed Time	43 minutes
Number of Rows Processed	4.5 billion
Number of Rows Inserted at Target	4.5 billion
Gigabytes Processed	340 GB
Gigabytes Per Hour	474.8 GB / hr
Rows Per Second	1,746,216
Time to Develop the ETL Job	6 minutes

结论

这些测试结果证明BO DI和Data Services能够满足绝大部分真实的大规模ETL工作要求。这些测试表明DI能够毫无困难地快速处理非常大的数据行数和数据量（大小），实现硬件平台允许的最大性能。

同样有趣的一个事实是，BO DI实现这样的性能并不需要进行采用什么特别的性能优化措施，这意味着开发人员在相同的硬件环境下实现相同的性能并不需要特别的技能和长时间的优化。开发这些高性能的Job仅仅花费了14分钟。

BO DI具备一些高级特性，来满足高端企业ETL性能需求，而不需要高级的技能来实现。

附录 1: 测试细节

源和目标

我们的场景是使用 TPC-C 基准测试数据库的模型作为数据源。TPC-C 是一个业界标准的交易处理基准测试，用来模拟具有大数据量的一个典型的订单录入系统。

TPC-Data Model

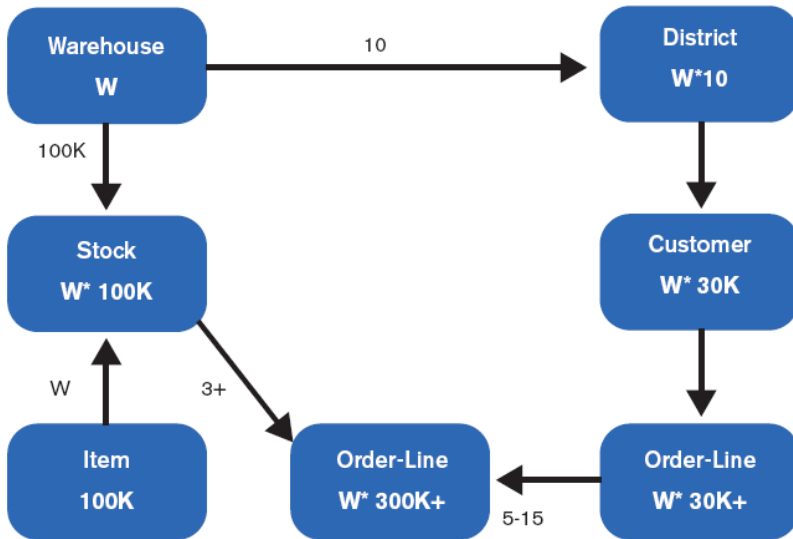


Figure 1: TPC-C Data Model [Source: tbc.ora]

数据源中的数据量如表 4:

The data volumes in the source database are shown in Table 4.

Table 4. Data Volumes in the Source Database		
Source Table	Number of Rows	Size in GB
Customer	450 million	269 GB
Item	100,000	10 MB
District	150,000	20 MB
Order	450 million	91 GB
Order Line	4.5 billion	249 GB
Warehouse	15,000	1 MB
Stock	1.5 billion	404 GB

目标模型是一个标准的星型模型，由 5 个维表和一个用来存储所有交易细节订单数据的事实表构成。目标表中的数据量如表 5 所示：

Table 5. Data Volumes in the Target Database			
Target Table	Sourced From	Number of Rows	Size
Dim_Customer	Customer	450 million	285 GB
Dim_Product	Item and Stock	100,000	10 MB
Dim_District	District	150,000	20 MB
Dim_Warehouse	Warehouse	15,000	1 MB
Dim_Time	Generated	365,000	10 MB
Fact_Order	Order and Order Line	4.5 billion	348 GB

所有的数据均需要经过处理加载到目标数据库中，目标表中的数据量主要由事实表和两个大维表构成。大维表拥有大量的数据行业有大量的数据列。

配置

我们使用了DI XIR2 (11.7) 进行本次测试，而使用DI XI 3则可预见能够获得相同或更好的性能。

数据源被设计成能够最快速度地向 DI 提供数据，以便消除由于数据源服务器造成的可能的性能瓶颈。数据源服务器配置如表 6：

Table 6. Source Database Server Specifications: Sun X4500	
Number of CPUs	2 x 2.8 GHz dual core
Memory	16 GB
Network Connection	10 GBit Ethernet
I/O	1 x 4GBit fiber channel
Storage	22 drives

目标数据库也设计成了无瓶颈系统，但同时也要测试真实的加载。为了让测试更简单，使用Oracle的ASM (Automatic Storage Management) 来管理所有数据库对象和存储，因此数据存储平台也不需额外进行优化。

目标数据库的配置见表 7：

Table 7. Database and ETL Server Specifications: Sun E25000	
Number of CPUs	72 x 1.8 GHz dual core
Memory	576 GB
Network Connection	10 GBit Ethernet
I/O	4 x 4GBit fiber channel
Storage	64 drives