ETL（extract, transform and load）产品乍看起来似乎并不起眼，单就此项技术本身而言，几乎也没什么特别深奥之处，但是在实际项目中，却常常在这个环节耗费太多的人力，而在后续的维护工作中，更是往往让人伤透脑筋。之所以出现这种状况，恰恰与项目初期没有正确估计ETL工作、没有认真考虑其**工具**支撑有很大关系。

做ETL产品的选型，仍然需要从以前说的四点（即成本、人员经验、案例和技术支持）来考量。在此，主要列举三种**主流**ETL产品：Ascential公司的Datastage、Informatica公司的Powercenter、 NCR Teradata公司的ETL Automation。

其中，ETL Automation相对其他两种有些特别之处，放在后面评述。

旗鼓相当：Datastage与Powercenter

就Datastage和Powercenter而言，这两者目前占据了国内市场绝大部分的份额，在成本上看水平相当，虽然市面上还有诸如Business Objects公司的Data Integrator、Cognos公司的DecisionStream，但尚属星星之火，未成燎原之势。

     谈Datastage和Powercenter，如果有人说这个就是比那个好，那听者就要小心一点了。在这种情况下有两种可能：他或者是其中一个厂商的员工，或者就是在某个产品上有很多经验而在另一产品上经验缺乏的开发者。为什么得出这一结论？一个很简单的事实是，从网络上大家对它们的讨论和争执来看，基本上是各有千秋，都有着相当数量的成功案例和实施高手。确实，**工具**是死的，人才是活的。在两大ETL**工具**技术的比对上，可以从对ETL流程的支持、对元数据的支持、对数据质量的支持、维护的方便性、定制开发功能的支持等方面考虑。

      一个项目中，从数据源到最终目标表，多则上百个ETL过程，少则也有十几个。这些过程之间的依赖关系、出错控制以及恢复的流程处理，都是**工具**需要重点考虑。在这一方面，Datastage的早期版本对流程就缺乏考虑，而在6版本则加入Job Sequence的特性，可以将Job、shell脚本用流程图的方式表示出来，依赖关系、串行或是并行都可以一目了然，就直观多了。Powercenter有Workflow的概念，也同样可以将Session串联起来，这和Datastage Sequence大同小异。

      ETL的元数据包括数据源、目标数据的结构、转换规则以及过程的依赖关系等。在这方面，Datastage和Powercenter从功能上看可谓不分伯仲，只是后者的元数据更加开放，存放在关系数据库中，可以很容易被访问。此外，这两个厂家又同时提供专门的元数据管理**工具**，Ascential有Metastage，而Informatica拥有Superglue。你看，就不给你全部功能，变着法子从你口袋里面多掏点钱。

     数据质量方面，两种产品都采用同样的策略——独立出ETL产品之外，另外有专门的数据质量管理产品。例如和Datastage配套用的有ProfileStage和QualityStage，而Informatica最近也索性收购了原先OEM的数据质量管理产品FirstLogic。而在它们的ETL产品中，只是在Job或是Session前后留下接口，所谓前过程、后过程，虽然不是专为数据质量预留的接口，不过至少可以利用它外挂一些数据质量控制的模块。

     在具体实现上看，Datastage通过Job实现一个ETL过程，运行时可以通过指定不同参数运行多个实例。Powercenter通过Mapping表示一个ETL过程，运行时为Session，绑定了具体的物理数据文件或表。在修改维护上，这两个**工具**都是提供图形化界面。这样的好处是直观、傻瓜式的；不好的地方就是改动还是比较费事（特别是批量化的修改）。

    定制开发方面，两者都提供抽取、转换插件的定制，但笔者认为，Datastage的定制开发性要比Powercenter要强那么一点点。因为Datastage至少还内嵌一种类BASIC语言，可以写一段批处理程序来增加灵活性，而Powercenter似乎还缺乏这类机制。另外从参数控制上，虽然两者的参数传递都是比较混乱的，但Datastage至少可以对每个job设定参数，并且可以job内部引用这个参数名；而Powercenter显得就有些偷懒，参数放在一个参数文件中，理论上的确可以灵活控制参数，但这个灵活性需要你自己更新文件中的参数值（例如日期更新）。另外，Powercenter还不能在mapping或session中引用参数名，这一点就让人恼火。

     总起来看，Datastage和Powercenter可谓旗鼓相当，在国内也都有足够的支持能力，Datastage在2005年被IBM收购之后，可以说后劲十足。而Informatica则朝着BI全解决方案提供商方向发展，Powercenter显然还将是它的核心产品。

   独树一帜：Teradata的ETL Automation

继续要说的第三种产品是Teradata的ETL Automation。之所以拿它单独来说是因为它和前面两种产品的体系架构都不太一样。与其说它是ETL**工具**，不如说是提供了一套ETL框架。它没有将注意力放在如何处理“转换”这个环节上，而是利用Teradata数据库本身的并行处理能力，用SQL语句来做数据转换的工作，其重点是提供对ETL流程的支持，包括前后依赖、执行和监控等。

这样的设计和Datastage、Powercenter风格迥异，后两者给人的印象是具有灵活的图形化界面，开发者可以傻瓜式处理ETL工作，它们一般都拥有非常多的“转换”组件，例如聚集汇总、缓慢变化维的转换。而对于Teradata的ETL Automation，有人说它其实应该叫做ELT，即装载是在转换之前的。的确，如果依赖数据库的能力去处理转换，恐怕只能是ELT，因为转换只能在数据库内部进行。从这个角度看，Automation对数据库的依赖不小，似乎是一种不灵活的设计。也正是这个原因，考虑它的成本就不单单是ETL产品的成本了。

    其实，在购买现成的**工具**之外，还有自己从头开发ETL程序的。

ETL工作看起来并不复杂，特别是在数据量小、没有什么转换逻辑的时候，自己开发似乎非常节省成本。的确，**主流**的ETL**工具**价格不菲，动辄几十万；而从头开发无非就是费点人力而已，可以控制。至于性能，人大多是相信自己的，认为自己开发出来的东西知根知底，至少这些程序可以完全由自己控制。

     就目前自主开发的ETL程序而言，有人用c语言编写，有人用存储过程，还有人用各种语言混杂开发，程序之间各自独立。这很危险，虽然能够让开发者过足编码的瘾，却根本不存在架构。

     有位银行的朋友，他们几年前上的数据仓库系统，就是集成商自己用c语言专门为他们的项目开发的。单从性能上看似乎还不赖，然而一两年下来，项目组成员风雨飘零，早已物是人非，只有那套程序还在那里；而且，按照国内目前的软件工程惯例，程序注释和文档是不全或者是不一致的，这样的程序已经对日常业务造成很大阻碍。最近，他们已经开始考虑使用ETL**工具**重新改造了。

|  |  |  |
| --- | --- | --- |
| **对比项** | **Informatica PowerCenter** | **IBM Datastage** |
| **产品完整性对比** | Ø  数据整合部分：PowerCenter，是业界公认领导者  Ø  数据质量管理：Data Quality，成熟稳定技术，在中国有大规模应用的成功案例。  Ø  实时数据捕获：PowerExchange，业界领先实时采集技术，支持广泛数据源的CDC和Realtime，与PowerCenter无缝集成。  Ø  元数据管理：Metadata Manager，是业界领先的企业级元数据管理平台，可做到字段级的元数据各项分析，有广泛的元数据采集接口，图形化无需编程，并可自动维护变更。 | Ø  数据整合部分：Datastage，属于业界一类产品  Ø  数据质量管理：QualityStage，收购的技术，不是主要其主要产品组成  Ø  实时数据捕获：MQ和DataMirror的技术，技术复杂，与DataStage是不同风格产品，产品的耦合度极差。  Ø  元数据管理：MetaStage，几乎免费的产品，应用性极差，并不能管理企业级的元数据。而新推出的产品与旧有产品线耦合度差，并未经过市场的考验。 |
| **开发人员的使用效率** | Ø  Informatica 是全图形化的开发模式，**不需要编码**，**工具**易使用，界面友好、直观。  Ø  专业的三天培训，可使开发人员快速入门，进行开发设计。  Ø  开发人员只要懂得数据库知识，即可。  Ø  Informatica 产品是以元数据为核心的，其开发过程中，所有的元数据，包括规则和过程，均是可复用，共享的。  Ø  经过简单配置即可支持大数据量的处理。  Ø  Informatica是完全基于引擎级别的，所有功能模块化，扩展性强，维护成本低。 | Ø  虽然也是图形化的界面，但复杂的转换过程，里面嵌入了很多类Basic脚本的成份。  Ø  要求开发人员，有编程语言基础。  Ø  在处理大数据量，必须使用Datastage企业版。但如果客户原先使用的Datastage 标准版，其作业的版本移植问题很大。这两个版本的工作平台、机制完全不同。作业移植，大概要有70%左右需要重新开发定义。  Ø  Datastage是基于脚本级的，底层基于PICK BASIC和COBOL（Main Frame上）内核开发，要求不同的平台需要不同的系统环境变量配置。 |
| **应用需求的改变和拓展的支持** | Ø  Informatica 是以元数据为核心的平台，现在完全支持SOA的思想，其最大特点就是完全支持松耦合．可拆分成Service 进行调用．这样需求变化，其需改动的部分，其影响会很小。  Ø  开发转换过程，均为共享的、可复用的。  Ø  元数据发生变化，可通过View Dependencies 功能，生成所有相关对象的报表，方便跟踪、校验，以应对需求的变化。  Ø  应用需求变化，调整作业后，直接可以运行，不需要重新编译。  Ø  作业移植等，也不需要重新编译。与平台和数据库无关。  Ø  支持跨操作系统的集群技术，可方便的进行平台级的扩展。 | Ø  需求发生变化，需调整相应的作业。如果是复杂需求，改动已有的脚本，其维护成本相对比较高。  Ø  每次作业变化调整，均需重新编译，才可执行。  Ø  Datastage企业版与Datastage 标准版，其作业的版本移植问题很大。这两个版本的工作平台、机制完全不同。作业移植，大概要有**70%左右需要重新开发定义**。一旦新的需求，需要企业版，其移植和再次开发，工作量要增加很多。  Ø  也因为两个版本的不兼容和脚本编译的开发模式，使之产品面对变化和扩展上，均有一定的限制。 |
| **项目实施的支持** | Ø  Informatica结合15多年的数据集成领域的经验，总结出一套针对Informatica产品实施数据仓库、数据管理等项目的最佳方法论Velocity 2008。该成熟的开发方法论，是指导客户实现快速、高质量项目实施的最佳武器。  Ø  现在全国拥有众多的名高级技术专家与顾问，与国内如大唐，联创、神州数码、东软，中软等多家知名集成商成立战略合作伙伴，Informatica产品开发人员全国上千人规模。  Ø  Informatica支持服务中心是有非常熟练的技术支持工程师充当的，这些工程师具备你需要的、成功的专家知识。在中国有专门的售后服务工程师。 | Ø  无专业/成熟，基于产品的项目最佳开发方法论  Ø  很难找到熟悉类Basic开发语言的Datastage开发工程师  Ø  IBM是以服务为主的公司，如果客户采用了其DataStage产品，将要支付大笔的IBM咨询服务费。 |
| **产品安装** | 完全图形化安装，无需额外安装平台软件，且不需修改系统内核参数 | Ø  需耗用时间安装和准备C编译环境，不同平台软件安装的C编译器也不尽相同  Ø  **需修改系统内核参数，对其他应用影响较大，有潜在的危险**。 |
| **产品升级** | Ø  平滑升级，完全图形化，不需修改已设计完作业。  Ø  主要是升级资料库，工作量很小。 | Ø  需重新编译已有作业  Ø  大版本之间以及跨平台的升级，很多作业需重新编写/编译代码，重复操作和维护工作量大。 |
| **产品移植** | Ø  PowerCenter**支持逻辑和物理设计分离的开发模式**，有一个Mapping（逻辑的）和Session（物理的或者可运行）的概念，Mapping是逻辑上的ETL规则，而Session才是真正可以实例化运行的任务。  Ø  可以跨平台、跨不同数据库进行作业的单个、整体移植。不需改变作业设计等，原有的任务可以直接在新环境下运行，并且只要更改Session的数据库联接串，则使用原有的Session任务访问不同的数据库类型数据，大大简化项目移植的工作。 | Ø  如果数据源，目标类型变化了，得修改以前所有的Job。  Ø  必须在新平台上编译所有作业，此移植的工作量较大。 |
| **元数据开放性** | Ø  元数据资料库可基于所有**主流**系统平台的关系型数据库(Oracle、DB2、Sybase ASE，Microsoft SQL Server)  Ø  用户可通过Informatica提供的Metadata Exchange视图读取元数据，Informatica提供详细文档描述其元数据视图的结构和含义。  Ø  可通过CMW标准跟其它BI**工具**共享元数据。  Ø  提供Global元数据库，Local资料库可共享Gloal资料库的元数据。  Ø  元数据对象可以快速导出/导入为XML文件，快速在多个项目中共享元数据  Ø  通过PowerCenter的Administration Console，PowerCenter元数据库可快速的导出为Informatica能识别的二进制文件，该元数据内容可再快速移植到其他关系型数据库中。 | Ø  DataStage的元数据DB是基于Universe7的，Universe不是一个开放型的数据库，（Universe目前最新版本是Universe10），IBM采用的元数据库技术上已明显落后。 虽然有计划要将之实现DB2的支持，但从本质上改动产品，不是短期内可以实现的。  Ø  **元数据库没有log备份恢复机制，如果知识库损坏后很难修复，易导致整个项目丢失。**  Ø  元数据库需要特定的Client才可以查看，所有DataStage在Universe中的元数据没有相关的描述，  Ø  **难与其它元数据库交互共享元数据，跨不同环境的元数据共享，必须重新编译该对象。**  Ø  无Global元数据库 |
| **元数据管理** | Ø  **Metadata Manager**可整合包括模型**工具**、数据集成、数据库、报表**工具**的各环节元数据的综合平台，支持各种**主流**UNIX平台和window平台。  Ø  元数据管理可跨不同**工具**平台进行血缘分析，在表级和字段级均可自动匹配。生成血缘分析和影响分析报告。  Ø  **Metadata Manager**提供跨各种**工具**的元数据血统分析；提供自定义元数据接口。  Ø  **Metadata Manager**支持OMG的CWM标准，其XConnects的接口非常广泛，扩展性高。 | Ø  MetaStage 是其元数据管理**工具**，仅能运行在 NT平台上.  Ø  可进行简单的元数据分析报告。且元数据分析只在表级，**不能进行字段级元数据分析**。  Ø  不能跨**工具**进行元数据跟踪，无法完成整体连贯的血缘分析和影响分析。 |
| **操作便捷性** | Ø  全图化开发，无编码，操作性强  Ø  被TDWI连续10年评为“数据仓库最佳实践”奖 | Ø  脚本式**工具**，需要学习类Basic语言  Ø  需要写大量的类Basic脚本，不便于快速开发以及后期维护。  Ø  增加了开发周期和投资成本 |
| **健壮的安全性** | Ø  多范围的用户角色和操作权限（只读、操作和设计等）  Ø  提供基于Folder的权限管理  Ø  权限可以分到用户或组  Ø  使用细致的锁（Lock）机制，提供了完善的安全，便于多用户的协作开发  Ø  可基于LDAP认证模式 | Ø  只提供少量角色：Product Manager，Developer和 Operator  Ø  只提供基于数据库的权限  Ø  没有基于Folder的权限管理，开发者只要有Developer的权限即可访问所有的Folder。  Ø  元数据的安全性和完整性没有很好的保障措施 |
| **并行处理** | Ø  可并行多个Session提高性能  Ø  Session支持多线程和管道技术(pipeline）  Ø  支持Session分区功能，性能跟CPU数据可达到基于线性的增长。  Ø  可将Session的分区任务分发到多个节点上(Session on Grid功能)，性能可随着节点的增加而增长。  Ø  Informatica支持跨不同操作系统的GRID。 | Ø  不能很好的多个Job的并行  Ø  并行组件(Torrent)需另付费购买，在Window平台上，无法使用Torrent的并行技术。  Ø  IBM DataStage企业版中才提供Torrent Orchestrate并行功能，由于在ETL设计时需要考虑很多Torrent Orchestrate的技术因素，并且没有具体的技术文档，开发难度较大。 |
| **单个任务的并行能力** | Ø  提供多种Session分区(Partition)功能  n  Round-Robin  n  Hash Auto-Keys  n  Hash User Keys  n  Key Range  n  Pass-Through  n  Database Partitioning  Ø  可将单个Session分配给 Grid，PowerCenter会根据系统资源和用户配置，将单个Session任务拆分为多个子任务，以达到多并行任务的负载均衡。并且随着Grid中的可用资源(Node)的增加，该Session的性能理论上会得到不断的提高。 | Ø  必须使用Datastage企业版才可以实现单个任务并行功能  Ø  **若想实现并行，必须要在源和目标数据库上各安装一套Datastage企业版，增加了企业的投资成本。**  Ø  **而且标准版与企业版兼容性极差，作业需要重新开发。** |
| **运行灵活性** | Ø  可在mapping环节点上，执行Pre sql 和post sql；可执行SQL 语句块  Ø  在session的点上，执行pre-session cmd、Post-session success cmd、Post-session failure cmd、On success E-mail、On failure E-mail等命令  Ø  对于非连接lookup，可实现动态调用  Ø  Mapping的源/目标可通过参数控制。 | Ø  无pre sql、post sql和session cmd 的功能  Ø  Lookup方法一：使用ODBC来做一行行的搜索，速度很慢  Ø  Lookup方法二：使用自定义的hash file来索引要Lookup的表，该文件存放在Universe数据库中。  Ø  **Lookup操作的hash file需要维护和调优，其性能也不稳定，经常在没有提示的情况下崩溃。** |
| **读取文件列表(File List)功能** | Ø  可灵活读取多个同结构/非同目录的多个文本文件。  Ø  可快速读取大数据量的File List | Ø  无内置的读取文件列表(File List)的功能，开发和维护量较大。 |
| **大数据量操作的性能** | Ø  有多种任务并行以及Session分区的技术  Ø  有官方TPC-H性能测试报告，PowerCenter 8.1在64位Linux平台上的性能达到了**7.7MB/sec/CPU。**  Ø  大数据量处理性能稳定。  Ø  **ETL性能可跟硬件资源(CPU为主，内存为辅)达到近线性增长！** | Ø  无官方性能测试报告  Ø  **大数据量的汇总、排序、关联，Lookup效率差且不稳定，作业会莫名其妙的死掉。** |
| **跨广域网/防火墙的安全数据传输** | Ø  PowerChannel可实现跨广域网和防火墙，实现加密/压缩的安全数据库传输。 | Ø  没有产品和方案实现该功能 |
| **对异常数据的处理** | Ø  提供Session级别的“**Row Error Logging**”功能，可自动捕获异常数据，能将所有出错的记录写到关系型数据库表中，包括如下主要信息：  n  出错记录的Row ID  n  出错时记录的当前值  n  出错记录的原始值  n  出错代码和信息 | Ø  无内置捕获错误记录的功能  Ø  必须写大量的脚本，读取其log来得到错误记录信息。 |
| **任务的自动恢复(Recovery)** | Ø  **自动Workflow Recovery功能(需要HA选项)**：短暂的异常(如网络故障)会暂时停止Workflow的运行，故障被修复后，Informatica可从Workflow的断点处继续执行任务。  Ø  **自动Session Recovery功能：**Informatica能保存Session失败前的断点信息，可从断点处继续运行任务。 | Ø  没有断点恢复的能力 |
| **Mapping**  **开发模板** | Ø  Informatica提供了PowerCenter Mapping Architect for Visio功能，能在VISIO中开发Mapping模板，快速产生多个Mapping，大大提高多个类似任务的开发和维护。在数据迁移，数据仓库的项目当中非常有用。  Ø  提供了Mapping Wizards，内置了很多的Mapping开发模板，如数据仓库中经常用到的Slowly Changing Dimensions 3种类型的模板，从而可提高开发效率。 | Ø  无缓慢变化维向导，需自行开发，开发效率低。 |
| **ETL任务调度** | Ø  提供Workflow Schedule功能，可连续调度任务  Ø  可基于时间/自定义事件/支持文件来触发任务  Ø  可实现实时的数据抽取 | Ø  不支持连续运行（continuously）和频率执行调度，无法实现数据达到即时抽取的能力，频率执行需通过自编译内部代码或循环工作流人工逻辑实现，开发难度和维护难度较高。 这是DataStage设计开发中一个比较棘手的问题。 |