在iUAP DI中提高数据库性能

在典型的数据仓库场景下，使用iUAP DI可以从各种数据库读取和写入数据。在大多数情况下，都是使用数据库的JDBC驱动完成这个工作。所以在读写数据库操作时都要通过中间的JDBC抽象层，下面我们分析一下读写数据库的详细过程。

如果步骤中指定了数据库连接，通常是选择一个数据库连接名。这样步骤就可以找到数据库的元信息。在转换运行时，在步骤的初始化阶段会建立数据库连接。如果所有使用数据库的步骤都可以成功的建立数据库连接，转换就开始运行，这时就可以向数据库中读写数据。需要说明的是每个步骤都会建立一个独立的数据库连接。

下面从DI对数据的操作和数据库本身两方面优化阐述：

1. 处理数据的往返操作

当SQL语句和参数值被传到数据库，数据库就会返回执行SQL语句，并返回结果。这样就产生了一个往返过程。所以和数据相关的DI步骤严重依赖这种往返操作的速度。而这种往返操作速度取决与网络速度、网络延迟和数据库性能。

1. 网络速度

网络的速度也就是带宽，会影响性能，尤其有很多数据需要通过网络来处理时，网络的带宽就更加重要了。象“表输出”和“插入更新”步骤加载数据库，要通过网络交换大量的数据。

你可能对网络的性能无能为力。但是，可以在加载数据之前，把数据移动到和数据库尽可能近的地方。“一辆装满硬盘在高速公路上奔驰的大货车的带宽是无可比拟的”。换句话说，对于一些慢速网路，可以使用一些非常规手段，比如把数据放到一个磁盘和或光盘上，使用快递邮送过来。

1. 网络延迟

运行DI的计算机和数据库服务器，网路距离越远网络延迟越大，导致性能差异就会巨大。对于网络延迟有时无能为力，可以试试其他解决的办法。首先，看能否减少对数据库操作的往返次数，例如：在“数据库查询”步骤中使用缓存可以有效的减少对数据库的操作。其次，使用批量处理，在“表输出”中选中”使用批量插入“选项，把数据行放到大的批量提交数据块里，可以减少数据库的往返操作次数。

1. 关系型数据库的操作

数据库本身的性能对转换的性能也起着重要作用，下面是一些通用的调优规则：

1. 批量提交大小

在支持批量提交的步骤中，可以根据使用数据库的不同，可以尽量将提交数设置的大些，但如果设置的过大就会占用过多的数据库的重做表空间，反而使性能会降低，一般来说”提交记录数量”设置为5000较为合适，如果数据性能较高且重做表空间设置也比较大可以设置更大些。

1. 索引

在做查询和数据表连接时，表中引入索引是至关重要的。在”数据库查询“和”维度更新“中应对关键字段增加索引。但索引会使数据库的的更新、插入、删除操作较慢。在关系型数据库中创建的索引越多，这些操作就会越慢，因为这些原因，在做大规模操作时，如数据初始化，批量更新等操作，可以先删除索引，然后再做完这些大规模操作后再创建索引，会更节省时间。

1. 表分区

对于大表数据可以考虑使用数据表分区，分区可以按照时间，也可以按照Hash值等，但无论分区按照哪种逻辑，都应该让数据库能快速知道一条记录在哪个分区里。这样在使用索引前，就会所限查询范围，相应提高查询性能。

1. 约束和触发器

数据仓库中有约束和触发器会降低数据库的性能，因为每次DML操作都要验证约束和执行触发器，所以在数据仓库中要尽可能少使用约束，不要使用触发器。

1. 数据库日志

通常对关系型数据库的DML操作都会写日志，对于数据库仓库日志没有OLTP那么重要，可以考虑修改设置，使尽可能少输出日志。例如：在Oracle中通过alter table <table\_name> nologging;可以大幅减少该表的日志输出，从而提高效率。