在iUAP DI中提高读文本文件的性能

当从文本文件读取数据时，有一系列的操作可能是性能瓶颈。具体原因可能很多，为了深入调查原因，首先要弄清楚读取文件都做了哪些事情。

首先，读取文件的转换步骤要从磁盘中读取文件所在的数据块。如果从一个慢速的磁盘中读取，读取的性能肯定不高。另外磁盘的寻道时间也会影响磁盘的读取性能。如果要求“CSV输入”或 “文本输入”频繁读取较小的数据文件或读取较小的数据块，那么就会花费一定的磁头重定向时间。

当文件读取数据后，DI会把读取的字节转换成字符，这些字符就构成的文本或字符串，把字节转换成字符使用的编码必须和文件原来的编码相同，否则会乱码。使用不同的编码方式的转换效率是不同的，例如使用双字节UTF-16和四字节的UTF-32是非常占用CPU的。

在获得字符后，就要把这些字符分割成不同的字段。“CSV文件输入”和“文本输入”选用格式是CSV时，会根据回车换行符读取每一行文本，然后把这行文本按照分隔符分割成若干字段。如果“文本输入”的格式使用的Fixed，则根据字符数分割字段。

最后，抽取的字段要转化成对应的数据类型，如Date，Number等。日期类型和数值类型的转换比较消耗CPU。

以上这些都依赖计算机的处理性能，尤其是读取很多字段做数据类型转换时，通常消耗大量的CPU时间。除此之外，还有trim操作，使用默认值替换空值等。

通过以上分析读取文本的过程，可以从以下的技巧来加快读取文本文件的速度。

1. **读取文本文件时使用延迟转换**

字符编码和数据类型的转换导致的CPU消耗似乎是无法避免的。但在一些场合下，使用“CSV输入”的“延迟转换”选项。如下图：



如果选中了该选项，所有关于数据相关的转换，如字符编码、数据类型转换、trim操作等，都会尽可能的延迟。输入步骤只是以二进制方式读取文件，并分成若干字段。

很明显，后面的步骤如果需要读取这些字段数据，数据转换还是会进行，但在以下的情况使用延迟转换可以提高性能。

* 如果大多数字段是以简单地以同样的格式写入另外一个文本文件。
* “CSV输入”后接“Oracle加载”。Oracle加载工具直接读取文本，使用延迟转换会加快生成临时文本文件的速度。
1. **配置NIO的大小**

“NIO缓存大小”这个参数对性能也有影响，可以在“CSV输入”中设置该参数。如下图：



这个参数决定了从文件中一次读取数据块大小。没有一个明确的规则来指导该设置多大合适。如果设置的比较大，每次读取的时间过长，会影响转换并发。一般来说，如果磁盘的缓存较大或寻道时间较短，可以把缓存设置小些，这样的运行可能会快。

1. **改变文件位置**

如果文件存放在FTP或Windows网络共享上，则文本文件的读取不仅要依赖远程磁盘的读取速度还依赖网络的传输速度。如果文本文件读取存在性能瓶颈可以考虑改变文件的存放文件位置，把文件存放到读取性能较高的磁盘上可以显著提高读取文件性能。

总之，文本文件的读取的性能瓶颈在磁盘的读取和类型的转换上，前者依赖磁盘IO速度，后者依赖CPU速度。合理设置文件位置和缓存大小可以提高读取的速度。减少不必要的类型转换可以减少CPU消耗，提高读取文本速度。