

ArcSDE 性能优化系列

之 ArcSDE 参数篇

题记

该文档主要是分析一下 ArcSDE 的相关参数对 ArcSDE 性能的影响,提起 ArcSDE 的性能,估计是 ArcGIS 用户非常关心的问题,性能无外乎查询性能、编辑性能的,本文档主要是以 RASTERBUFSIZE、MINBUFSIZE、MAXBUFSIZE 等一下缓冲区参数来分析这些参数的意义以及对性能的影响。

参数

这些参数可以在 SDEHOME/etc/geomgr.def 文件获得,也可以在 sdeconfig 命令获得,也可以直接进入数据库表 Server_Config 表中查看。

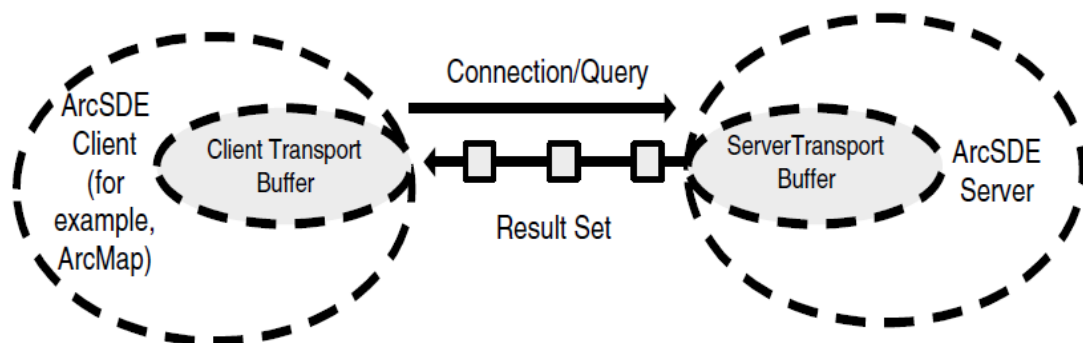
缓冲区大小初始化参数

ArcSDE 提供三个传输缓冲区初始化参数: MINBUFSIZE、MAXBUFSIZE 和 MINBUFOBJECTS。

数据通过数据流在 ArcSDE 服务(应用程序服务器)和客户端应用程序之间传输。数据流是用于查询表或图层的通信机制。将存储器结构传递给基础网络堆栈之前,数据流先将存储器结构(传输缓冲区)用作容器。

创建 ArcSDE 数据流时, ArcSDE 进程将传输缓冲区分配给客户端和服务端。传输缓冲区可以累积矢量数据记录,然后将这些记录批量(而非以单个记录形式)发送到整个网络,从而减少输入/输出(I/O)并提高性能。

这些记录收集在 ArcSDE 应用程序服务器进程的传输缓冲区内,并会在应用程序查询数据库时被发送到 ArcSDE 客户端传输缓冲区。或者,这些记录会收集在客户端的传输缓冲区内,并在应用程序将数据写入数据库时被发送到 ArcSDE 服务器进程的传输缓冲区。



传输缓冲区参数

这些参数共同构成了可防止应客户端的请求而传输缓冲区的阈值下限。检索查询请求的数据时，达到 MINBUFOBJECTS 或 MINBUFSIZE 后才会发送缓冲区内成批的数据。一旦所有已提取的数据都被置于缓冲区，即使尚未超出这些阈值，这些数据也将被传输到客户端。首先检查 MINBUFOBJECTS。

MINBUFOBJECTS 是传输缓冲区被发送到整个数据流之前存储在传输缓冲区内最小记录数。默认值是 512 条记录。该值也是最小值。ESRI 测试已证明，将该参数增大到 512 以上不会带来任何性能优势。

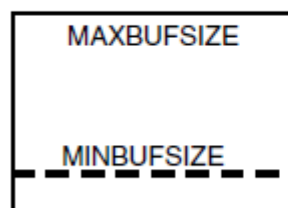
MINBUFSIZE 不应大于 MAXBUFSIZE 的二分之一。因为将 MINBUFSIZE 设置得过高可能会延长传输缓冲区前客户端的等待时间。如果 MAXBUFSIZE 设为 64 千字节、MINBUFSIZE 设为 56 千字节，那么客户端需要等到达到 56 千字节阈值时，才会传输缓冲区发送。最小值为 4,096 字节。

MAXBUFSIZE 参数表示分配到各传输缓冲区的内存总量。达到 MAXBUFSIZE 后，传输缓冲区将停止累积记录，然后等待将这些记录发送到客户端的请求。增加缓冲区的大小之前，请验证服务器上的物理内存是否充足，否则可能因为页面调度过多而降低性能。例如，默认值分配给每个缓冲区 64 千字节。如果有 100 个用户建立了连接，则每个 gsrvr 进程将包含一个共消耗 6.4 MB 系统内存的缓冲区。如果 MAXBUFSIZE 加倍，则会消耗 12.8 MB 内存。

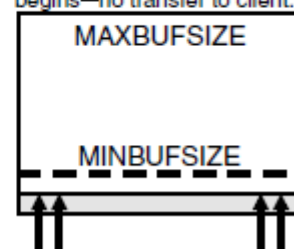
数据传输次序

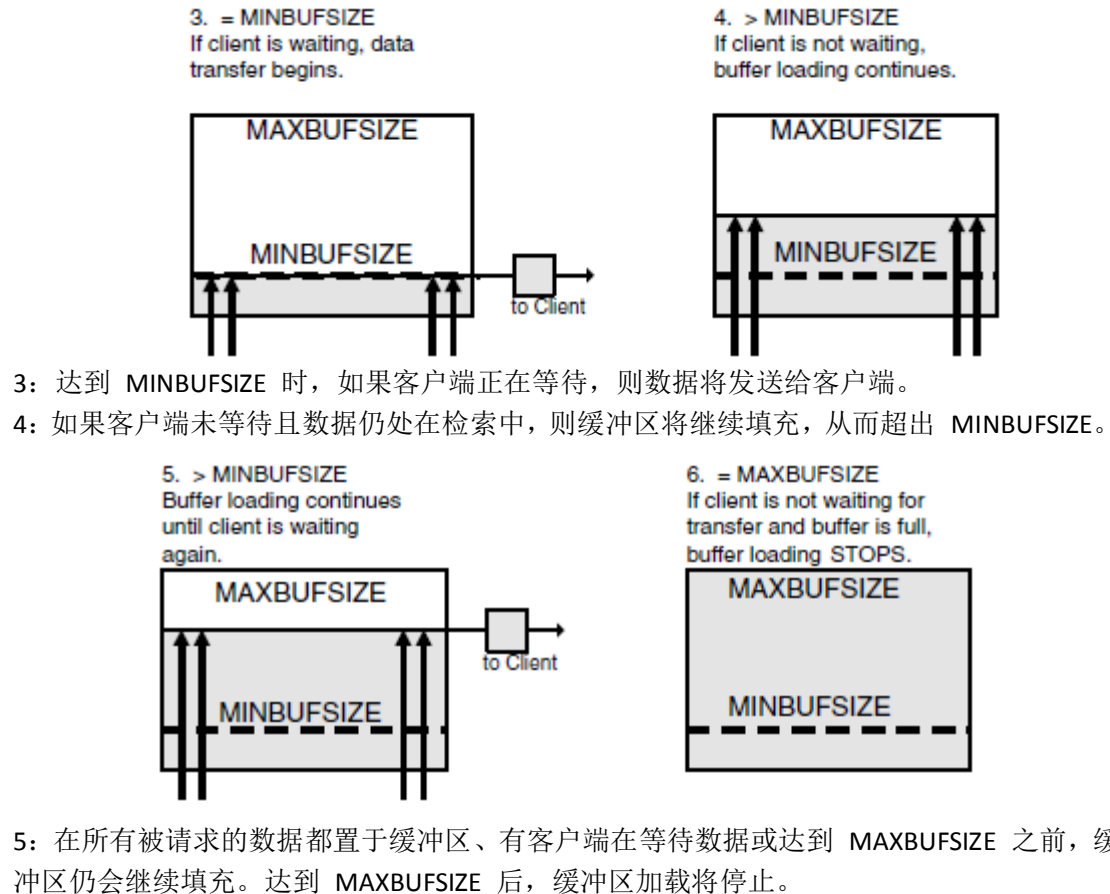
为响应查询而提取数据时，ArcSDE 将检查数据批中的行数是否满足 MINBUFOBJECTS 参数的要求。如果查询仍在处理之中，当达到 MINBUFOBJECTS 阈值（存在所需行数）并且客户端在等待数据时，数据被发送。如果未满足 MINBUFOBJECTS 的要求且数据仍在检索中，那么 ArcSDE 将检查数据的大小，以查明是否达到了 MINBUFSIZE。如果也未达到，则缓冲区将继续填充。

1. Transfer buffer empty.

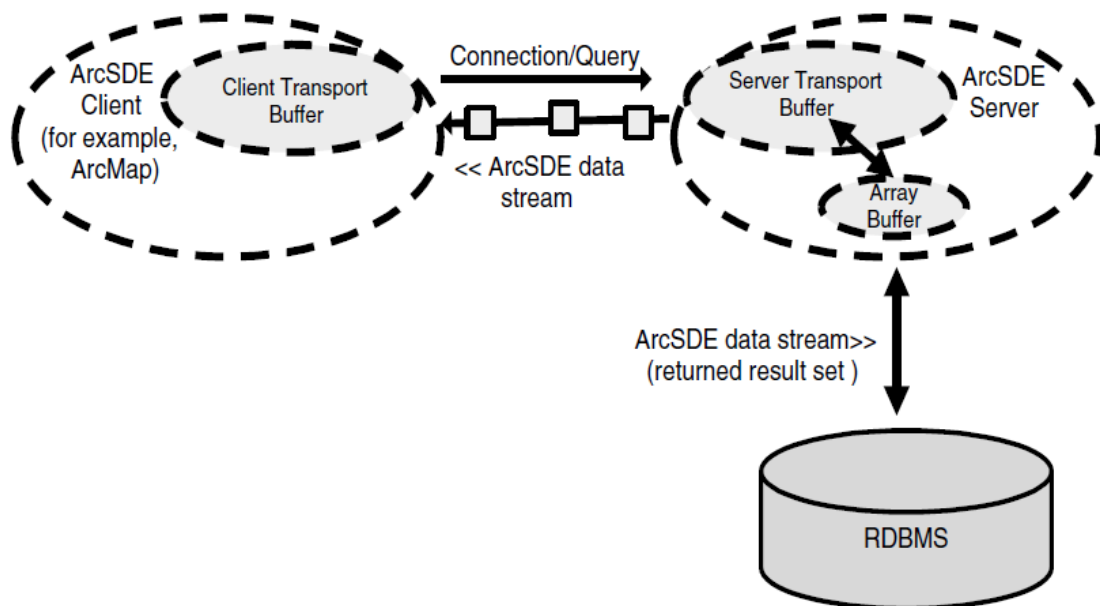


2. < MINBUFSIZE
Client sends query to server. Buffer loading begins—no transfer to client.





ARRAY 缓冲区参数



对每一个 ArcSDE 流的建立，ArcSDE 都会分配给 array buffer。当 ArcSDE 与 DBMS 进行数据的转换，例如对 DBMS 进行插入数据以及实施了 Array 操作，那么 ArcSDE 往 DBMS 插入记录也实行了 Array。所有支持的 Dbms，除了 Microsoft Access，已实施了 Array 读取机制，

所以每当中发出一条 select 语句的用户的名义读取或获取 Array 中的记录。

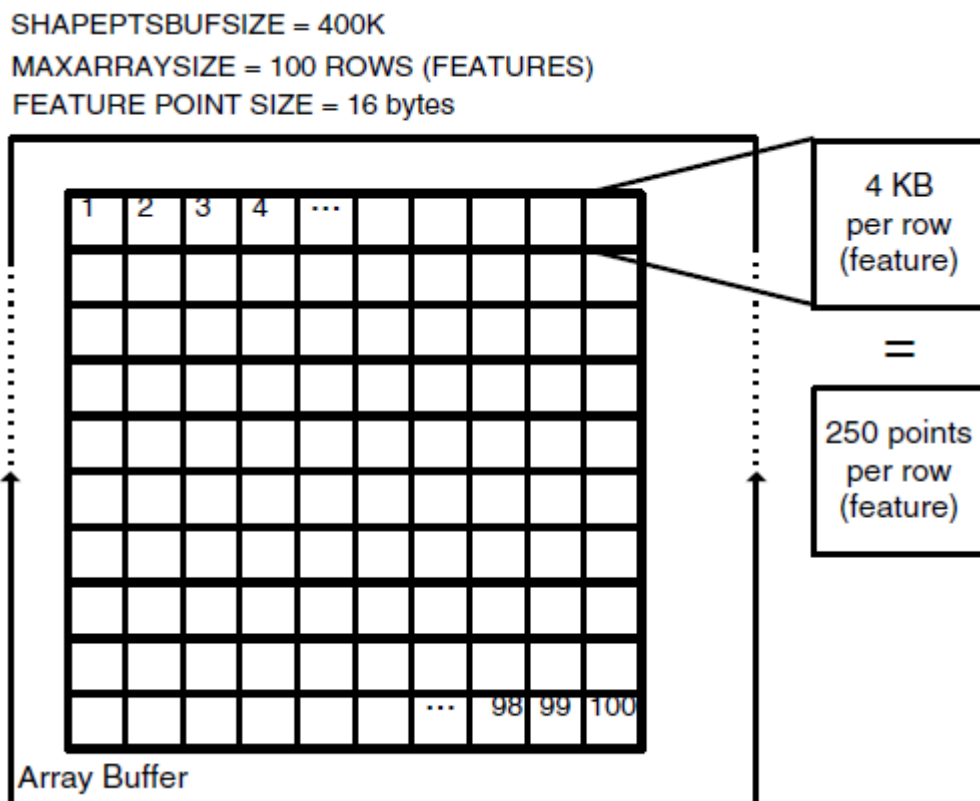
Array Buffer 减少了 ArcSDE 与 DBMS 服务器查询数据或者插入数据的 I/O, I/O 越小性能就越好, 但是如果设置过大的 Array Buffer 值也会导致过度分页, 一般来说, 大约 100 条记录是大多数应用程序可以处理 Array 中的最佳数目。

- MAXARRAYSIZE 100 #Max.array fetch size
- MAXARRAYBYTES 550000 #Max.array bytes allocated per stream
- SHAPEPTSBUFFSIZE 400000 #Shape POINTS array buffer size
- ATTRBUFSIZE 50000 #Attribute array buffer size

估算 SHAPEPTSBUFFSIZE

在进行 Array 插入的几何要素对象包含两部分: 要素的元数据和点数据。要素的元数据包括要素 ID, 点数, 实体类型, 要素存储空间, 该空间要素存储点数据, 另一方面就是要素的实际点数据。

SHAPEPTSBUFFSIZE 确定点的缓冲区大小数据。默认设置基于数组大小为 100。调整 SHAPEPTSBUFFSIZE 的最优设置是关键性能。ArcSDE 评估中的所有要素的平均大小基于 Array 大小 (MAXARRAYSIZE) 和大小点缓冲区 (SHAPEPTSBUFFSIZE)。如果超过了要素的平均大小, 它被标记为分开提取。



例如, 假定 SHAPEPTSBUFFSIZE 为 400000 (400K), 每次获取的 array 大小 (MAXARRAYSIZE) 为 100 行, 该要素不包含注记、Z 值、M 值, 那么每个点存储按 16Bytes (X 值为 8Bytes, Y 值为 8Bytes, 因为他们都是 Double 型的)

每个要素点的 array buffer 的最大空间=SHAPEPTSBUFFSIZE/MAXARRAYSIZE

400000 Bytes/100=4000 Bytes

进一步划分的最大的空间, 可用于每个要素返回每个点需要的字节的数的总数点, 可存

储的最大的空间，可用于在每个要素。

$$4000/16=250$$

上面的例子，ArcSDE 希望每个要素都不超过 250 个点（不管什么类型的要素都是以点坐标来存储），当 ArcSDE 遇到某个要素包含 251 或者更多的点，那么这个要素就会分开来提取也就增加了 DBMS 的 I/O 了。

假如要素包含 Z 值或者 M 值，那么就除于 24（8+8+8）或者全部包含，那么就除于 32（8+8+8+8）。

设置 SHAPEPTSBUFFSIZE 太小的话会增加 I/O，降低性能，如果设置的太大会浪费内存。

注意：

ATTRBUFSIZE 和 SHAPEPTSBUFFSIZE 的总和必须小于或等于 MAXARRAYBYTES。否则，将无法启动 ArcSDE 服务。如果出现这一问题，请增大 MAXARRAYBYTES，或者减小 ATTRBUFSIZE 或 SHAPEPTSBUFFSIZE。

该值不能使用 SE_connection_set_stream_spec 函数进行更改，它只能由 ArcSDE 管理员在 SERVER_CONFIG 表中进行更改。

那么根据上面的分析，我们就可以根据我们的数据来估算相应的值了。

```
C:\Users\gis>sdelay -o stats -l dltb,shape -u test -p test
```

ArcSDE 10.0 for Oracle11g Build 775 Fri Sep 17 11:45:27 2010

Layer Administration Utility

Statistics for Layer 161

Total Nil Features:	0
Total Point Features:	0
Total MultiPoint Features:	0
Total Line Features:	0
Total MultiLine Features:	0
Total SimpleLine Features:	0
Total MultiSimpleLine Features:	0
Total Area Features:	2210
Total MultiArea Features:	0

Total Features: 2210

Minimum Feature Number:	1
Maximum Feature Number:	2210
Largest Feature:	1570 Points
Smallest Feature:	4 Points
Average Feature:	10.40 Points

Minimum Polygon Area:	0.25161 sq.
Maximum Polygon Area:	128887.79904 sq.
Average Polygon Area:	647.42037 sq.
Minimum Polygon Perimeter:	2.00833
Maximum Polygon Perimeter:	19124.78636
Average Polygon Perimeter:	107.33751

Layer Envelope:

minx: 496658.33150, miny: 3639834.15680
maxx: 498252.06750, maxy: 3641141.39040

下面说说根据上面所得到的图层行数信息，单个要素的最多点信息来估算 SHAPESTBUFSIZE 值。

由上面可以得到 FeatureCount=2210, MaxFeaturePoints=1570，那么我们需要将最大点数 $1570 \times 0.75 = 1177.5$ ，然后使用以下 SQL

```
SQL> select count(*) from dl1b ss where ss.shape.numpts<1177.5;

COUNT(*)
-----
2208
```

如果得到的数目与总数的比例为 90%左右即可，如果不是或多或少，那么 0.75 就相应的减少或者增大，也就是说，尽量获得该图层 90%的要素点数，那么根据这个点数来获得 $\text{SHAPESTBUFSIZE} = \text{MAXARRAYSIZE} \times (\text{NUM OF POINTS}) \times (\text{Point SIZE}) = 100 \times 1177 \times 16$

影像数据导入参数

在 ArcGIS9 软件中使用 RasterDataset 或者 RasterCatalog 导入海量影像数据，一般都是要花费 N 长时间来完成数据导入工作。

该参数用于定义栅格缓冲区的大小，而栅格缓冲区的大小控制着栅格数据的转换。栅格转换包括数组缓冲区和传输缓冲区。栅格数组缓冲区被设置为 RASTERBUFSIZE 参数的两倍，而栅格传输缓冲区被设置为 RASTERBUFSIZE。因此，分配给服务器上的栅格转换的内存是 RASTERBUFSIZE 的三倍。

在客户端，当通过流访问栅格分块时，系统会将 RASTERBUFSIZE 字节的内存分配给客户端栅格传输缓冲区。直到流被关闭时才会取消分配栅格缓冲区，除非流被添加到流池。（请参见 STREAMPOOLSIZE。）

该值以字节为单位指定并且必须足够大以便能存储所访问的最大栅格分块。如果栅格分块不适用于传输缓冲区，则会返回 SE_RASTERBUFFER_TOO_SMALL 错误。

如果内存比较紧张，请指定一个更小的分块大小而不要增大 RASTERBUFSIZE。

计算每个像素的字节数时必须考虑像素深度。

示例：图像像素深度=64 位，分块大小=128 * 128，所需的 RASTERBUFSIZE=131,072 字节 (128 * 128 * 8)；图像像素深度=8 位，分块大小=256 * 256，所需的 RASTERBUFSIZE=65,536 字节 (256 * 256 * 1)

对于 ArcSDE 9.2 及更高版本，ArcSDE 栅格数据的吞吐量会被编入线程以更好地利用服务器的处理能力。此外，还会使用多个内存缓冲区。**因此，将 RASTERBUFSIZE 参数设置为不大于 1 MB 是非常重要的。**

在数据导入之前，我们可以将 RASTERBUFSIZE 的值调大来增大导入性能，提高效率。数据导入完毕再将数据修改为原值即可。

ArcSDE 连接数

这个就不做详细介绍了，估计这方面用户比较了解，主要是增大 ArcSDE 的连接数，然

后清除无效连接。

CONNECTIONS 在 Windows 默认为 48，在非 Windows 为 64，该连接数包括服务连接数和直连数，修改完毕后需要重启 ArcSDE 服务即可。

TCPKEEPALIVE 默认设置为 FALSE，这个需要将该值设置为 TRUE，清除一些无效的 ArcSDE 连接，这样就增强了并发的性能。但是该清除无效连接对直连没有效果，只针对服务连接。

说明

该文档为作者在学习 ArcGIS 的一些笔记（自己学习、ArcGIS 帮助、网络资源、高手请教相结合），难免有一些表达不全面或者理解有出入的，还请各位多多指教！

广告博客

<http://www.gisall.com/index.php?uid-121719> : 欢迎访问 GIS 博客

<http://blog.csdn.net/linghe301> : 欢迎访问 GIS 博客

或者在百度文科上搜索：linghe301

By Lish

