元数据读取代价建模与比较

元数据排列方式

在读取一个 TimeSeries 时，一般需要先读取 TimeSeriesMetadata（TSM），再读取若干个 ChunkMetadata（称之为 ChunkMetadataList，CML）。元数据目前在 TsFile 中的排布方式如图 1

图 1: TsFile 中的元数据排列方式

其中 TSM 存在元数据索引树的叶子节点中，每个 TSM 对应的 CML 则存在文件的另一个位置。因此，要读取一个 TimeSeries 时，首先要读取对应 TSM，然后经过一次寻道找到对应的 CML 的位置，将其读取。

一种新的排布方式如图 2

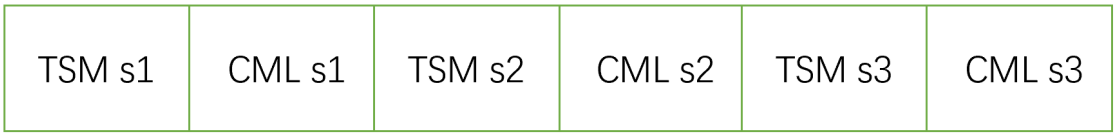

图 2: 一种新的元数据排布方式

在这一种新的排布方式中，每个 TSM 后面紧接着存储了对应的 CML，它们共同被存放在元数据索引树的叶子节点中。当读取一个 TimeSeries 时，找到对应的 TSM，读取完以后不需要经过寻道就可以直接读取对应的 CML。这样的排布可以减少一次寻道，但是有可能会使得索引树的叶子节点增多（因为索引树的叶子节点大小固定）。

符号标记

- S_{leaf}：叶子节点的大小，为 100 KB，其等于树的阶数乘以 S_{TSM}，当前系统中树的阶数为 1024。
- S_{TSM}：TimeSeriesMetadata 的大小，为 100 Byte。
- M：一个 ChunkMetadataList 里面 ChunkMetadata 的个数。
- S_{CML}：ChunkMetadataList 的大小，为 $M \times 80$ Byte。
- $Seek_{TimeSeriesMetadata}$：寻道到 TimeSeriesMetadata 的代价，设定长度为 100 KB，根据经验值估计为 0.53 ms。
- $Seek_{CMLToCML}$ 从一个 ChunkMetadataList 寻道到另一个 ChunkMetadataList 的代价，假设为 10 ms。
- $Cost_{read}(x)$：磁盘读取长度为 x 的数据的读取代价，由线性插值得到。

服务器 192.168.130.18 上对磁盘读取数据的测试结果如下：

数据段长度 / Byte	102400 (100KB)	104857600 (1MB)	3040870 (2.9 MB)	6606028 (6.3 MB)	9542041 (9.1 MB)	104857600 (100 MB)
平均读取时长 / ms	10.15	15.67	48.46	68.55	70.74	507.55

以上测试结果均为 100 次测试的平均值。

经过推导，两种元数据排布方式在不同场景下的元数据读取代价为

	原始排布方式	新排布方式
单序列聚合查询	$Seek_{TimeSeriesMetadata} + Cost_{read}(M)$	$Seek_{TimeSeriesMetadata} + Cost_{read}(M)$
N 个序列的聚合查询	$\lceil \frac{S_{TSM} \times N}{S_{leaf}} \rceil \times (Seek_{TimeSeriesMetadata} + Cost_{read}(S_{leaf}))$	$\lceil \frac{(S_{TSM}+S_{CML}) \times N}{S_{leaf}} \rceil \times (Seek_{TimeSeriesMetadata} + Cost_{read}(S_{leaf}))$
单序列原始数据查询	$Seek_{TimeSeriesMetadata} + Seek_{CMLToCML} + Cost_{read}(S_{leaf}) + Cost_{read}(S_{CML})$	$Seek_{TimeSeriesMetadata} + Cost_{read}(S_{leaf})$
N 个序列的原始数据查询	$\lceil \frac{S_{TSM} \times N}{S_{leaf}} \rceil \times (Seek_{TimeSeriesMetadata} + Cost_{read}(S_{leaf})) + N \times (Seek_{CMLToCML} + Cost_{read}(S_{CML}))$	$\lceil \frac{N \times (S_{TSM}+S_{CML})}{S_{leaf}} \rceil \times (Seek_{TimeSeriesMetadata} + Cost_{read}(S_{leaf}))$

N 个序列的原始数据查询

原始元数据分布

$Cost_{read}(S_{leaf})$ 通过经验值知为 10.15 ms。代价函数中得到

$$
\begin{aligned}
&\lceil\frac{S_{TSM}\times N}{S_{leaf}}\rceil\times(Seek_{TimeSeriesMetadata}+Cost_{read}(S_{leaf}))+N\times(Seek_{CMLToCML}+Cost_{read}(S_{CML}))\\
=&\lceil\frac{100B\times N}{1024*100B}\rceil\times(0.53ms+10.15ms)+N\times(10ms+Cost_{read}(M\times 80B))\\
=&\lceil\frac{N}{1024}\rceil\times 10.68ms+N\times(10ms+Cost_{read}(M\times 80B))\\
\approx&\ 0.0104N\ ms+10N\ ms+N\times Cost_{read}(M\times 80B)\\
=&\ 10.0104N\ ms+N\times Cost_{read}(M\times 80B)
\end{aligned}
$$

取近似 $Cost_{read}(M\times 80B)\approx M\times Cost_{read}(80B)=0.00793M\ ms$，则有

$$
原式\approx(10.0104N+0.00793MN)\ ms
$$

新元数据分布

$$
\begin{aligned}
&\lceil\frac{N\times(S_{TSM}+S_{CML})}{S_{leaf}}\rceil\times(Seek_{TimeSeriesMetadata}+Cost_{read}(S_{leaf}))\\
=&\lceil\frac{N\times(100B+M\times 80B)}{1024\times 100B}\rceil\times(0.53ms+10.15ms)\\
=&\lceil\frac{N+0.8NM}{1024}\rceil\times 10.68ms\\
\approx&\ 0.0104N\ ms+0.00834NM\ ms
\end{aligned}
$$

比较

原始元数据查询和新元数据查询的差值为

$$
\begin{aligned}
\Delta_{Cost}&=(10.0104N+0.00793MN)-(0.0104N+0.00834NM)\\
&=10N-0.0041MN\\
&=N\times(10-0.00041M)
\end{aligned}
$$

因此有

$$
sign(\Delta_{Cost})=\begin{cases}+, & M<24390\\ -, & M\geq 24390\end{cases}
$$

即当一个 CML 里面 ChunkMetadata 的个数小于 24390 时，新排布好于原始排布；当大于等于 24390 时，原始排布好于新排布。