

## BigStream 机制详解

由来:

在早期版本的 ZServer4D 中, SendStreamCMD 这类机制(所有使用到 DataFrameEngine 打包的机制), 都是针对小型 Stream 的打包和编码。它们处理大型 Stream, 都是拆包来干, 假如收发一个 1G 的文件, 会有 2000 个小型 SendStreamCMD 命令用于分包, 在早期, 处理大型 Stream 是十分麻烦的。

后来, 工程开发和应用需要一种大型的 Stream 处理, 为了区别于 SendStreamCMD, 就定了 BigStream 概念。它与 SendStreamCMD 的差异就是针对大型 Stream。

BigStream 在 ZServer4D 可以工作与 MemoryStream 也可以是 FileStream, 也可以是其余自定义的 Stream。

后来, 又经过了几个月的应用模型研究, 针对数据库批次操作的 BatchStream, 被开发出来, 它完全构建于 BigStream 的基础上, 以 MemoryStream 作为暂存, 按批次性的存储, 我们一次性提交大规模的修改用 BatchStream 是非常方便的。

机制:

BigStream 不会对 Stream 本体做编码解码, 它只会给 Stream 加一个数据头, 说明 Stream 的长度和命令前缀符。

BigStream 在发送的过程中, 兼容超大型文件机制, 收发的尺寸可以到 TB 级。每次发送和接收 BigStream 命令都会是一个碎片, 由你来决定如何处理这些碎片, 你可以用 FileStream 来追写碎片, 也可以用 memoryStream, 通过对碎片叠加追写的检查, 接受到的 Size 只要达到预期就表明一个 BigStream 已经发送完成。这一概念和早期的手动处理碎片是一致的, 只不过, 今天的 BigStream 是自动化的触发碎片接收事件。

BigStream 和 BatchStream 的差异:

BatchStream 完全依赖于 BigStream 来工作

BigStream 每次接收触发的碎片事件需要手动来处理, BatchStream 会自动化的组合 BigStream 的碎片, 并且会将接受完成的 BigStream 放到一个以 MemoryStream 为媒介的暂存区, 以供使用

BatchStream 不适合大型 Stream 操作

注意: BatchStream 必须在双通道系统下才能工作, BigStream 在单通道也能工作

BigStream 和 CompleteBuffer 的差异 :

BigStream 每次收发的碎片需要手工来处理, CompleteBuffer 会自动化的组合碎片

BigStream 每次发送前都会拆分大型数据, 以碎片形式发送和接收。CompleteBuffer 不会拆分数据, 直接发送

BigStream 是针对 Stream 媒介做数据处理, CompleteBuffer 是针对内存块做数据处理

注意: BigStream 性能远远低于 CompleteBuffer

By600585

2018-9