有关 mmap 的基准测试

Elasticsearch/Lucene 使用 memory-mapped file 技术实现大文件的高性能读写。

和普通读写文件比较，主要特点是：

- 避免多次重复的内存复制，从 user space 到 kernel space
- 数据对齐，减少多余数据的复制
- 数据回写磁盘一般由 os 执行，因此即使应用进程崩溃，之前写入内存的数据依然会正常处理
- 数据已经写入内存，但电源故障或 os 崩溃可能造成文件丢数据
- 适合高效操作大文件，即使文件比内存大很多
- 在内存不足时可能频繁 page fault，造成性能下降

考察 mmap 的特性，积累指标基准测试数据，是掌握 Elasticsearch 性能特点的关键，因为 mmap 的基准数据是 Elasticsearch 的性能上限之一。

Java

Lucene 使用 Java 编写，因此最基本的基准测试需用 Java 编写。

最简单 mmap 基准测试

基本步骤如下：

1. 编写一个 spring boot 项目
    - 一个 web app，不必做任何事情，只需根目录可以访问显示 hello world，说明 app 正常工作即可
    - 让这个项目支持 docker 构建，参考Spring Boot with Docker
2. 使用 visualvm 查看 spring boot 项目运行中的 JVM 信息
    - visualvm 地址：https://visualvm.github.io/index.html
    - 下载、运行，有中文介绍
    - 可以通过 JMX 方式连接到 docker 容器中的 spring boot web app
    - 参见：visualVm 通过 jmx 连接 docker 中 Springboot 项目
    - 正常使用后，找到显示 `non-heap memory` 的地方（mmap 使用的内存区域）

3. 在 spring boot 项目中加入 mmap 相关代码
    - 为了在 spring boot web app 加载后自动加载 mmap 相关代码
        - 不讲究的话，用 `static {}` 静态初始化块
        - 或者 `ServletContextListener`
    - 代码主要内容
        - 使用 `RandomAccessFile`，一个上下文路径，比如'./test.log'
        - `MappedByteBuffer` 是 nio 包中实现 mmap 的类
        - 执行循环，向 MappedByteBuffer 中加入字节，循环到比如 30MB
        - 循环结束后，关闭 MappedByteBuffer，删除文件
        - 重新开始新的循环，循环往复
        - 这部分代码的目的是让 mmap 相关内存不断变化，便于后续监控识别
4. 在 visualvm 中监控到项目的 mmap 内存变化指标