LSM-Tree Phase2 报告

2025 年 4 月 13 日

1 背景介绍

在现代存储系统中，简单的键值对插入、删除、查询功能并不能满足所有需求。例如，搜索引擎、电商、视频软件推荐系统、图像处理系统等，都需要“语义相似度搜索”功能，而不能只使用精确的匹配功能。

上一阶段中的 LSM-tree 实现了基本的插入、删除、查找功能。这一阶段中，为 LSM-tree 添加了 search_knn 函数，实现了语义检索功能，将值转化为向量，通过余弦相似度计算向量的相似程度，并获取与目标词最相似的字符串。

本实验测试了这一功能的各个部分所用时间。

2 测试

检索功能可大致分为五个部分：维护向量、获取维护的向量、计算问题向量、计算余弦相似度、排序，其中维护向量和计算问题向量是与大模型相关的主要部分。本实验测量了五个部分各自用时占比。

此外，对于大模型中计算向量的函数 embedding，本实验将其分为文本预处理、初始化批处理、批处理嵌入计算三部分，测量各自用时。

2.1 实验设置

实验运行在 Linux 环境下。

首先，实验读入 trimmed_text 文件的前 120 行，将代码修改为维护向量和不维护向量的情况，分别测量用时并计算差值，减少计时的开销，并得到维护向量的大概用时。然后，进行 120 次 search_knn 操作，计算其中各部分用时，可以得到检索功能五个部分各自的用时占比。

此外，对于计算向量函数的三个部分，也进行 120 次操作，计算各个部分用时与占比。

计时使用 std::chrono 库。

2.2 预期结果

检索功能的五个部分中，维护向量和计算问题向量与大模型最相关，预计用时最多。

对于其它三个部分，获取向量需要遍历所有键值对，时间复杂度为 O(n)；计算余弦相似度复杂度也为 O(n)；排序时间复杂度为 O(nlogn)。所以，获取向量和计算余弦相似度所用时应该较短，在同一个数量级，排序时间略长。

在 embedding 函数中，初始化批处理不涉及大模型调用，文本预处理与大模型的分词器有关，而批处理嵌入计算涉及到多次大模型调用。

综上，检索功能五个部分预期用时时长为：维护向量 $\approx$ 计算问题向量 $>$ 排序 $>$ 计算余弦复杂度 $\approx$ 获取向量；embedding 函数三个部分预期用时时长为：批处理嵌入计算 $>$ 文本预处理 $>$ 初始化批处理。

2.3 实验结果与分析

实验得到检索功能五个部分用时如表 1。

操作	维护向量	获取向量	计算问题向量	计算相似度	排序
120 次操作用时	133861.67ms	1ms	121609.50ms	6ms	0
平均用时	1115.51ms	8.33μs	1013.41ms	50.00μs	0
用时占比	52.39%	0.0004%	47.60%	0.0023%	0

表 1: search_knn 操作五部分用时

可以看到，除排序外，实验结果与预期相同。实际用时为维护向量 $\approx$ 计算问题向量 $>$ 计算余弦复杂度 $\approx$ 获取向量 $>$ 排序。

在获取向量和计算余弦相似度时，不仅遍历了所有键值对，还遍历了“值”字符串中的每个字符，在字符串较长的情况下，时间复杂度不能再简单地计算为 O(n) 而应是 O(mn)，而排序是标准的 O(nlogn)，常数很小，所以出现了排序速度快于获取向量和计算余弦相似度的情况。

embedding 函数中，三部分各自用时如表 2。

操作	文本预处理	初始化批处理	批处理嵌入计算
120 次操作用时	8.67ms	2.67ms	91916.67ms
平均用时	72.25μs	22.25μs	765.97ms
用时占比	0.0094%	0.0029%	99.9877%

表 2: embedding 操作三部分用时

实验结果为批处理嵌入计算用时 $>$ 文本预处理用时 $>$ 初始化批处理用时，与预期相符。

在检索功能中，与大模型相关的耗时超过了 99.99%，更具体地，在 embedding 函数中，调用大模型耗时也超过了 99.99%，并且耗时与大模型调用次数相关。

3 结论

综上结论为，检索功能五个部分用时时长为：维护向量 ≈ 计算问题向量 > 计算余弦复杂度 ≈ 获取向量 > 排序；embedding 函数三个部分用时时长为：批处理嵌入计算 > 文本预处理 > 初始化批处理。

此外，在增加语义相似度搜索这一功能后，由于插入时要维护向量，导致 LSM-tree 的 put 操作慢了几个数量级。如果要对支持这种功能的系统进行优化，那么对大模型的优化是非常重要的。此外，也要尽可能减少大模型调用的次数。

4 致谢

感谢知乎、维基百科等博客、网站提供的参考；感谢 deepseek、kimi 等大模型提供的思路与帮助。

感谢提供支持的朋友们。

5 其他和建议

（对大模型不太感兴趣，写完之后总有一种学了但是没学的感觉，不过看到结果还是觉得很奇妙的）