moonseek 项目申报书

1、项目名称

moonseek —— 纯 MoonBit 全文搜索平台：库 / 服务 / 分布式集群

2、项目简介

moonseek 是一个完全用 MoonBit 编写的搜索平台：既是可嵌入的库，也是可独立运行的服务，还是可水平扩展的集群。纯 MoonBit 核心零 FFI，编译到 native / WebAssembly / JavaScript；一个二进制按参数即可作为单机节点、**分片**或**协调器**运行。能力涵盖倒排索引与 BM25、精确与松散短语、免词典中文分词、模糊纠错、分面与数值聚合、**HNSW 向量检索与 BM25×向量混合排序**、HTTP 检索服务、以及**一致性哈希分片、写入复制、跨分片聚合、在线扩容与副本修复**与全局一致评分、故障转移；**混合检索同样跨分片扇出，且与单机逐条一致。**已发布 mooncakes v0.17.5（355 项测试全绿、native `--deny-warn` 零告警；CI 十六项含质量回归、跨目标快照互通，以及集群 / 复制 / 扩容 / 修复 / 修复边界 / 分片重启 / 持久化 / 并发持久化 / 混载 soak **九份**端到端校验），并有在线演示：**浏览器里直接搜索 mooncakes 全部 3,698 个真实包。**

分布式正确性实测：3 分片 vs 单机同语料，`?dfs=true` 下 **8/8 查询排序逐位一致、最坏分数差 0.000e+0。扇出成本**：分片连接复用后，3 分片协调器查询 p50 均值 **0.94–1.02×**（与单机持平）、单跳转发写 **1.06×**、双成员复制写 **1.46×**（530→364 docs/s，全组确认）；复用前分别为 1.62× / 2.19× / 3.10×。**一致哈希**：3→4 片只搬 **24.2%** 文档（取模搬 74.9%，理论最优 25%）。**复制**：3 000 篇经协调器写入，6/6 成员落盘，杀主分片后结果逐字节不变；整组失效精确上报 `ok=2, failed=1`。**在线扩容**：预测搬 762 篇、实际搬 762 篇，四片分布与预测**逐项相同**，总量不丢不重，重跑为空操作。**副本修复**：掉线副本回来后 `_repair` 精确补齐 1007 篇，脏文档被删除，随后杀主分片答案不变。**跨分片混合检索**：5 个查询在 `knn_exact=true` 下命中 id 与词法/向量名次**全部相同（5/5）**，协调器复用引擎同一个 `fuse_rankings`。**Block-Max WAND**：20k 文档 top-10，单条饱和词跳过 **70–74%** posting（2.11→**1.01 ms**）；合取由最短词驱动（`search AND engine` 4.27→**2.71 ms**）；析取**带闸门**，仅稠密词 +稀 10 倍词时走并集 WAND（快 **29%**、跳 75%），其余留给精确归并，五种形状均有测试钉住；所有路径与全扫**逐字节一致**（严格小于才跳，平局仍评分）。**短语/前缀**：短语改「最稀词驱动+前向游标」，`"search engine"` 7.07→**0.29 ms（24×）**、松散 9.80→**1.04 ms**；单词项前缀走 block-max（`seg*` 2.34→**0.44 ms**），多词项最小堆归并。**服务端持久化**：`--data-dir` 每条确认写入先落 CRC journal（fsync）再进内存，重启后**文本/分面/数值/向量/删除逐条回来**；批量共享 fsync，持久化 ingest 仅 **1.2–1.3×** 内存模式；并发写走 group commit 共享 fsync，8 写者 **1.9–2.2×**、32 写者 **2.7–3.4×**；混载 soak 抓到并修掉一个真实丢数据窗口：checkpoint 可截断未 apply 的 journal 记录。**向量**：点缓冲按字段缓存（重建 4.29 ms 由每查询一次降为每变更一次）；HNSW vs 暴力 ef=32 达 0.996、ef=64 达 0.999。

检索质量可复现：`tools/eval` 在全量注册表上算 NDCG@10 / MRR@10 / Recall@20，产出 `docs/quality.md` 与消融表——字段加权 0.9045、等权重 0.8236、只搜包名 0.4082（加权比子串搜索高 122%）；长度归一在该语料为 −1.1%，报告如实写出。标注由语料结构派生，与检索器输出无关；报告声明其不可与 BEIR/MS MARCO 相比。

3、项目方向，通用性说明

方向：数据处理与数据库 / 基础工具库（填补 MoonBit 生态在全文检索基础设施上的空白）。

通用性：moonseek 不绑定任何应用场景或数据格式——检索能力以通用库 API 暴露（Index 建索引、Document 加文档、Query 查询语法、FacetFilter 过滤、Snapshot 序列化），字段模式由调用方声明。因此同一个库既能做服务端文档检索，也能做浏览器端离线搜索、本地知识库、代码检索或 RAG 召回底座；核心不依赖文件系统与网络（native 文件层单独可选），可被任意 MoonBit 项目作为依赖复用。

4、至少 3 个完整的预期使用场景

- **场景一：服务端文档检索（已落地）。**`moonseek-server` 可直接部署：批量灌文档、检索（分面过滤 / 数值聚合 / 高亮片段 / 打分解释）、拼写纠错、词项统计；索引以带 CRC 快照落盘，WAL 保证崩溃不丢数据。仓库内 `console/` 是驱动它的 Web 控制台。
- **场景二：浏览器端零后端搜索（已完整落地）。**`demo/` 索引 **mooncakes.io 上全部 3,698 个已发布包**的包名、关键词、描述与依赖边；索引由**原生二进制**构建为 1,398,936 字节（1.33 MiB）快照，再由**浏览器 wasm-gc** 直接加载——同一份字节跨两个后端。实测快照加载 ~200 ms、中位查询 **0.20 ms**、wasm 二进制 246,961 字节（241 KiB）；同页可拖入自有文件即时建索引，并持久化到 IndexedDB 供刷新后免解析恢复。
- **场景三：本地笔记与代码检索工具。**CLI 把笔记目录或代码库索引到本地快照，native 直接读写文件；写入先落 WAL，异常退出后重放恢复；索引损坏可检测并拒绝加载，而非静默返回错误结果。仓库内 `tools/registry` 就是这一路径的真实用例。
- **场景四：横向扩展（已落地）。**按 id 做**一致性哈希**分片（128 虚拟节点），协调器不持有数据，查询并行扇出后归并；写入按归属路由并**复制到该组每个成员**，因此副本真的能接管；需要与单机一致的排序时开两阶段全局统计，需要容错时配副本。缩容扩容前可用 `GET /_plan?shards=N` 按真实文档 id 算出要搬多少。扩容时先跑 `GET /_plan?shards=N` 拿到「要搬多少」，再 `POST /_rebalance?from=N` 原地搬（先写目的、全组确认后才删源）；副本掉线回来后 `POST /_repair` 与主对齐。可同时跑多个协调器（它们不持有状态）。CI 真起三片/三主三副/四片扩容，跑通「排序一致 / 聚合等于单机 / 复制后故障转移 / 扩容后仍等于单机 / 修复后接管无感」。
- **场景五（延伸）：AI 应用的检索底座。**BM25 与向量检索同进程融合（RRF 或加权，返回每条命中在两个榜单的名次），可先按字段/分面过滤缩小候选，再给出可解释的候选（逐词项 tf/df/idf 可分解），直接充当 RAG 第一级召回，无需外部向量库。

5、拟实现的核心功能

已完成：倒排索引与多字段加权；BM25（k1/b 可配）；CJK 二元切分；布尔 / 精确短语 / 松散短语（`"a b"~N`，最小跨度算法）/ 前缀 / 字段限定；模糊搜索（编辑距离 + BK-tree，2.4 万词项提速 17–491×）；拼写纠错；分面过滤与计数；数值聚合（terms / stats / histogram / percentiles）；高亮与片段；逐词项打分解释；排序控制与游标分页；段式索引；WAL 崩溃恢复；CRC 快照（MSK3）；HNSW 向量检索与混合融合；HTTP 检索服务与 Web 控制台；**一致性哈希环与再平衡测算**；**分布式分片 / 协调器 / 写入复制 / 故障转移**；**跨分片分面与聚合（与单机逐项一致）**；**跨分片排序（score/id/数值字段，18/18 与单机一致）**；**在线扩容执行器（`_plan` 预测 = `_rebalance` 实际）**；**副本修复（双向）**；**多协调器**；**跨分片混合检索（BM25×向量，与单机逐条一致）**；**Block-Max WAND（单条词与合取；跳 70–74% posting，与全扫逐字节一致）**；**一键演示脚本与端到端演示叙事（含分镜脚本）**；CLI 与浏览器演示。

6、是否为原创项目、移植项目或参考已有开源项目

原创项目，非移植。未复制任何第三方实现；算法（BM25、CJK bigram、BK-tree、WAL 帧格式、FNV-1a+fmix32 哈希、HNSW、RRF）为公开文献的独立实现，代码全部自行编写。

7、如为移植项目，需说明参考项目名称、来源链接和许可证

不适用（原创项目）。项目依赖仅两个，均为 Apache-2.0 且仅使用未修改的公开 API：`moonbitlang/core`（标准库）、`moonbitlang/async`（仅用于 native 层文件与网络）。项目自身以 Apache-2.0 开源。

8、GitHub 仓库链接

https://github.com/NaIWaniubi/moonseek

仓库公开，含 82 个 commit（每个对应一次完整功能/修复/文档变更，附实测输出与测试）。内含：双语 README、架构文档、性能白皮书、**检索质量报告与消融实验**（`docs/quality.md`）、**分布式设计文档**（`docs/cluster.md`，含哈希环/复制/聚合/混合检索/扇出成本五组实测表与「未实现清单」）、**HTTP 接口契约**（`docs/api.md`）、22 维特性矩阵（对比 tantivy/Lucene/bleve）、三层可运行示例、Web 控制台、**十六项 CI**（ubuntu×windows×三目标 + 基准 + 质量回归 + 集群 / 复制 / 扩容 / 修复 / 修复边界 / 分片重启 / 持久化 / 并发持久化 / 混载 soak 九份端到端）、可复现的语料构建 / 质量评测 / 集群对比 / 再平衡测算工具、独立评审与修复报告。性能白皮书与分布式文档逐条列出未测量项与对己不利的测量结果，而非只报喜。