**马鸣庸 微信：Mamio\_Johnson**

**邮箱**：mamiom7ma@gmail.com **电话**: 858-539-6919 **个人网站**: https://incandescent-licorice-a37843.netlify.app/

**学历背景**

|  |  |
| --- | --- |
| **2022.9-2023.12 加州大学圣地亚哥分校 (cs ranking 世界第三)**  **2018.9-2022.06 武汉大学** | **计算机科学与技术 硕士**  **遥感科学与技术 本科** |

**技能**

**编程语言：**Go, C++, Python, HTML, CSS, JavaScript, Java

**技术:** socket编程, 分布式系统(Raft), 数据库(B+ 树), Spark, 图像处理，时间序列分析

**项目经历**

**2023.03-2023.06. 缓冲区管理的B+树**

·在I/O层之上构建了一个**缓冲池**(Buffer Pool), 并实现了**缓冲替换策略**(Buffer Replacement Policy)和**LRU时钟算法**。

·在缓冲层之上构建了B+树, 支持INSERT/DELETE操作, 支持插入int，double和char\*变量, 并支持三层及以上结点。

·采用**批量加载**(Bulk Loading)初始化B+树，并在数量超过”**fanout**”时**分页**。

**2023.01-2023.03. 分布式云文件管理系统**

·实现了http协议(简洁版本):

· 实现了http持久链接(**HTTP persistent connection**): 客户端可以重用到给定服务器的TCP链接

· 提供安全控制(**safe control**)，不允许客户端访问文档根目录以外的内存。

·创建了一个名为SurfStore的容错云文件存储服务(**fault tolerant cloud-based file storage system**)，使用**gPRC**进行通讯

· SurfStore服务由两个部分构成:

· BlockStore: 存储这些块(**block**)，并在给定标识符时，检索并返回相应的块。

· MetaStore: 管理文件的元数据(**meta data**)和文件名到块的映射(通过**SHA-256**进行散列编排(marshal))。

· 客户端的文件数据在本地数据库中以版本(**version**)的进行存储。运行客户端时，会发生同步操作(sync)，并将新文件添加到基本目录后上传到云端，其他客户端同步到云端的文件将被下载到基本目录，解决任何有冲突的文件。

· 使用**一致性哈希算法**(Consistent Hashing Ring)在不同的BlockStore中存储和管理块。

· 通过**RAFT**协议确保MetaStore具有容错性(fault tolerant)，意思是即使在少数服务故障(minority of server failure)的前提下，始终保持一致性(consistent).

**2022.9-2022.12. 操作系统实现**

·实现了操作系统的线程结构：如**Alarm**()函数用来call定时器中断；**Join**()函数用来在子线程完成前令父线程保持睡眠；其他还有**Yield**(), **Fork**()等；实现了**Semaphores**用来保证原子性(**atomicity**).

·为每个用户进程创建pageTable数据结构, 该结构将进程的虚拟地址(**virtual address**)映射到物理地址(**physical address**).

·实现按需调页(**demand paging**),分页替换(**page replacement**)用以释放物理页面框架(physical page frame)处理**page fault**.

**工作经历**

**2023.6-2023.09 Adobe 软件开发实习生 美国硅谷**

·构建Adobe主要AI平台基础设施，将API调用从阻塞等待改为**同步非阻塞**，并将**网络I/O从**1GB减少到每次推理调用9MB。使用**Jmeter**进行负载测试，能够生成1600 TPS（每秒token数量）。

·使用**fastAPI**实现，能够对任务进行CRUD操作，并使用**alembic**版本控制将其保存在Postgres DB中。

·为每个fine tuning任务生成一个Docker容器，能够根据负载对容器数目进行调整。

·做了**FlanT5**和**llama2**模型的fine tuning工作，使用**PEFT**使得模型具有独立存储基础模型和微调层的能力。

**2022.7-2022.08 亚马逊软件开发实习生 深圳**

·开发了一个图像处理算法，该算法将深度学习与Unsharp算法相结合，达到了比平板电脑中使用的相机算法更低的锐度结果。并使用**MTF-50**评估算法的性能。

·通过使用**adb**控制成像设备并在亚马逊实验室中生成不同锐化(sharpness)参数的图像，用于比较算法性能。

·开发了一个用户友好的模型，可以手动或自动控制锐度(sharpness).

·在亚马逊实验室使用**Imatest**软件测试了包括**Sobel**，**Canny**算子和**Unsharp**算法在内的经典图像处理方法。

**2021.11-2022.02 联想数据分析实习生 北京**

·利用联想的历史销售数据以及来自其他公司(如IDC和GFK)的数据，进行时间序列预测(**time series forecast**)，预测联想笔记本和平板电脑的未来销售。

·通过实施**Prophet**等机器学习算法以及**LSTM**和**GRU**等深度学习模型，将模型的预测准度提高了4.2%。