**个人资料**

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 吴元昆 | 性别 | 男 |
| 出生日期 | 1986-7-1 | 工作年限 | 4年 |
| 手机 | 15810951365 | 电子邮件 | [wyk@mail.ustc.edu.cn](mailto:wyk@mail.ustc.edu.cn) |

**个人经历**

|  |  |  |
| --- | --- | --- |
| 2014.3——2016.7 | 高德地图 搜索技术研发部 | 资深研发工程师 |
| 2012.6——2014.3 | 盘古搜索 广告搜索技术研发部 | 高级研发工程师 |
| 2009.9——2012.5 | 中国科学技术大学 软件学院 | 硕士 |
| 2005.9——2009.7 | 华东交通大学 软件学院 | 本科 |

**技术能力**

|  |  |
| --- | --- |
| 技术基础 | 熟练掌握C/C++语言，熟悉python、shell语言  熟练掌握基础数据结构和算法，常用设计模式。  熟练掌握多线程并发、同步机制、网络接口编程。  熟悉Linux的基础系统环境。 |
| 技术专长 | 在线检索引擎的索引结构设计、性能优化。  在线搜索服务架构设计。 |

**技术项目经验**

|  |
| --- |
| 语音对话系统独立架构设计 (2016.3——2016.6)  角色: 架构设计成员 ＆ 项目开发组负责人  系统内部分四层层设计: 服务框架、对话状态控制层，业务策略层，数据适配层。  服务框架: 管理通信，并发容量控制，对外服务，及应用内基础组件管理。  对话状态控制层: 控制整体流程，调度语义状态模块对上文对话状态与当前交互语义做适配，决策场景-意图-状态的流转，调度具体业务组件处理任务。全景维护用户会话状态。  业务策略层中，每个独立的Policy组件根据自身场景-意图，生成具体任务，调度服务获取数据，生成槽位信息和话术。业务策略层采用插件化接口设计，适应多垂类产品的快速接入和低成本的升级迭代。  数据服务适配层隔离业务数据服务源，提供对不同业务数据源访问服务，并提供内部数据接口的适配。  目前，对话系统已支持指令类，地图poi类，导航类，调侃类，路况信息等业务。其他泛娱乐业务待接入。 |
| 搜索部模型字典查询引擎组件HSDT(2015.8——2015.11)  角色: 架构设计负责人 ＆ 项目开发负责人  字典查询组件分两部分设计: 字典托管平台和字典查询引擎。  字段托管平台: 集中透明的监控进程内所有模型字典的变更状态，采用串行事件流方式控制各字典的更新操作，防止内存尖峰。  字典查询引擎: 自适配最优散列表和映射算法，降低碰撞率。  引擎存储空间设计: 冲突由散列表迁移至数据段，有效简化压缩了散列表结构空间；数据段聚簇连续存储，冲突关系采用offset自适应压缩存储，有效提高了运行内存资源利用率，同时，减少了冲突数据的内存访问跳转次数，提高了cpu cache命中率。  字典引擎对亿级模型词典平均平均冲突比1.08，访问性能在1us级；二进制词典内存膨胀率平均在5%左右(无value去重)。  引擎提供可配置的value去重编码选项。模型数据业务方自行权衡配置。  同时，组件提供配套离线工具: 字典索引生成器和索引校验工具。  接口设计精简，组件已在搜索部全面推广: 业务调度模块，策略分析模块，语义分析模块，分词模块，纠错模块；搜索建议模块均以接入HSDT替换原阿里的tdbm组件。 |
| 地图检索引擎优化(2014.11——2015.5)  角色: 架构开发成员  地图检索引擎优化分三部分: 索引瘦身，查询过程优化，detail服务分离。  索引瘦身分两块设计: 倒排压缩采用原信息，PForDelta+分块，bitmap多方式的自适配最优压缩方法存储；正排信息字段精简规约，字段值排重编码，BitRecord定长字段打包等方法瘦身。索引加载空间瘦身近30%。  查询过程优化的实现: 倒排链分块+二分查找+目标块局部解析，索引暗序+渐进式查询。几步优化，使poi检索过程有效控制了召回深度，缩小召回结果规模(特别是高latency query情况)，且减少了中间结果存储，快速海选出高质量备选集合。使整个检索过程平均性能提升5倍，平均查询响应时间降至5ms。  detail服务分离独立: 有效减少了索引增量，且大幅缩短了实时展现数据的上线流时间。 |
| 搜索服务容灾系统(2014.7——2014.9)  角色: 架构开发成员  容灾系统分两部分设计: 容灾系统和在线服务过载保护框架。  容灾系统内部又分两部分设计: 泛化容灾cache和绝缘延迟环境。泛华容灾cache每天凌晨根据前一天日志进行线下泛华训练，产生cache数据，更新容灾cache。绝缘延迟环境则处于完全隔离状态，作为容灾cache的补充，降低灾难情况下的搜索无结果率。  过载保护框架: 根据进程内worker繁忙比例，分过载预警，过载保护，过载退场3个阶段，对服务单节点实施自我保护。 |
| 广告检索引擎(2012，盘古搜索)  角色: 开发成员  广告检索引擎分四层设计: 框架层，业务层，控制层，基础数据引擎层。  框架层: 承载对外服务和对内的并发负载调度。  业务层: 实现业务相关的请求解析/扩展，及结果渲染等。  控制层: 提供对基础数据层的召回，过滤等访问的功能接口，支持业务流程；及处理平台提交的增量更新请求，完成对基础数据索引的更新。  数据引擎层: 实现对索引数据的加载，多级表内存offset关联；及索引读取，更新等操作。同时对数据访问进行阶段划分，定长变长分段，同阶段数据打包合并存储，一次访问，整体取出。  广告搜索系统相比较与网页搜索系统: 广告主对购买词，竞价，预算，投放地域，投放时段的操控很多，统计值每天会有40%左右的广告计划会发生修改操作，引起索引更新。广告的召回截断相对轻量，过滤截断操作相对重。  数据索引对多值的正排过滤字段，转化为<Field，value>的0/1命中映射，对齐简历bitmap倒排，由此把正排过滤操作bitmap倒排过滤。两方面收益: 1)索引的原地更新，广告主的频繁修改操作，映射成为<Field，value>对应位的0/1置位操作，有效减少了索引更新带来的内存结构问题；2)bitmap的高效的过滤操作和连续内存访问集中特点，有效避免了普通正排取值判断的多次内存跳转访问，提高了cpu cache的命中率，提成了检索过滤截断的性能。 |