

异数操作系统 众筹

关键字: DPDK,C10K,C100M,TCP 协议栈,同步阻塞线程,协程,IM,IOT,消息推送,“零”运营。

异数（东邪）操作系统简历:

2002 年,一种基于 RTOS 理论的非抢占阻塞线程模型的实时操作系统。

2005 年,引入异步微内核思想,引入设计期线程原语 (DSL),简化线程业务逻辑设计。

2016 年,增加惰性资源 QOS 调度,使用阻塞线程模型重新设计 TCP 协议栈。

2014 年,放弃异步微内核思想,回到同步阻塞线程模型,引入业务层 IOPS QOS 控制。

异数操作系统特性:

编程模型:

1. 广泛扩展的同步阻塞线程模型。
2. Actor, CSP 编程模型体验。
3. 简单可设计的精细 QOS 调度器。

性能 (每核):

1. 5000W 线程生成能力。
2. 3000W Event 生成能力。
3. 1 亿线程 4G 内存占用。

惰性资源 QOS 调度:

1. 高度可用的系统资源。
2. “零”系统错误。
3. “零”软件层中间缓存。
4. “零”软件层传输缓存。
5. “零”高性能定时器依赖。
6. 高度自适应优化的 DPDK-ring PMD。

其他:

使用巨页对象池线程池技术,新建销毁 0 开销。

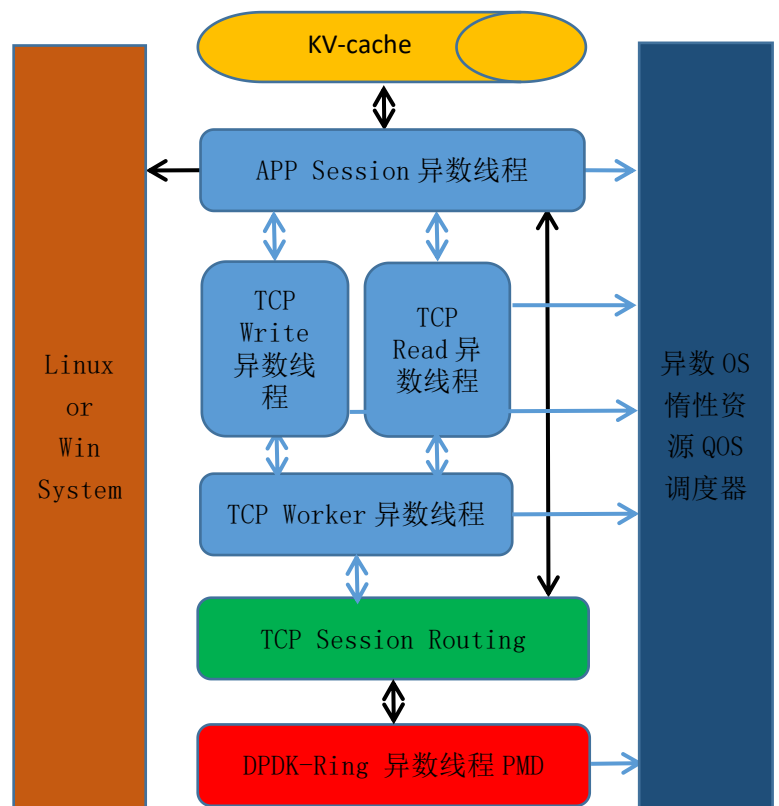
众筹合作方式:

4. 多合伙人基金,技术维护年金。
5. 买断。
6. 公益捐助,精神捐助,开源。

联系人:叶树深

联系方式:QQ 1055783080

电话:15026893648



异数 TCP 协议栈特性:

性能:

1. 每核 3M 新建能力, 10M 查询推送能力。
2. C100M, 30G 内存, 300 字节每 Session。

技术特性:

1. “零”中间缓存。
2. “零”传输缓存。
3. 惰性资源 QOS 代替的 TCP 拥塞控制算法。
4. IOT IM 领域优化的 C100M TCP 协议, 被抑制的阻塞窗口。

异数操作系统 众筹

异数（东邪）操作系统简介：

2002 年，一种基于 RTOS 理论的非抢占式实时操作系统，配合高性能定时器做调度，编程复杂，混合异步 IO 模型的同步逻辑，最终由于业务逻辑设计复杂而放弃。
教训：设计模型，设计语义很重要。

2016 年，增加惰性资源理论充实 QOS 调控，使用阻塞线程模型重新设计 TCP 协议栈，适配 DDPK RSS run-to-complete RDMA 等技术与模型。

2005 年，引入异步微内核思想，引入设计期线程原语（DSL），简化线程业务逻辑设计。
2008 年，设计去中心的资源分享系统（磁性连）由于异步线程逻辑表达复杂而失败。
2008 年，设计百万 NPC AI 系统（使用 UDP）由于延迟过高，错误无法处理失败。
2009 年，内网警察，海量 AI 监视网络可疑访问，清洗网络访问。
教训：异步队列的微内核操作系统，内部太多 DDOS，队列太深，延迟过高，错误处理代价高，只能用于不需要错误处理，不需要同步响应，全异步的业务。

2014 年，放弃异步微内核思想，采用同步阻塞模型，引入业务层 IOPS QOS 控制。
2015 年，设计一款 TCP 代理，做海量链接消息转发，1000W 链接 40WTPS，局限较大，宿主操作系统容易因为外在因素在协议栈发生雪崩，40WTPS 也不足以发挥万兆网卡性能，无法多核扩展。
教训：不抛弃传统 OS 协议栈是不行的，需要用海量阻塞线程模型重新设计 OS 协议栈

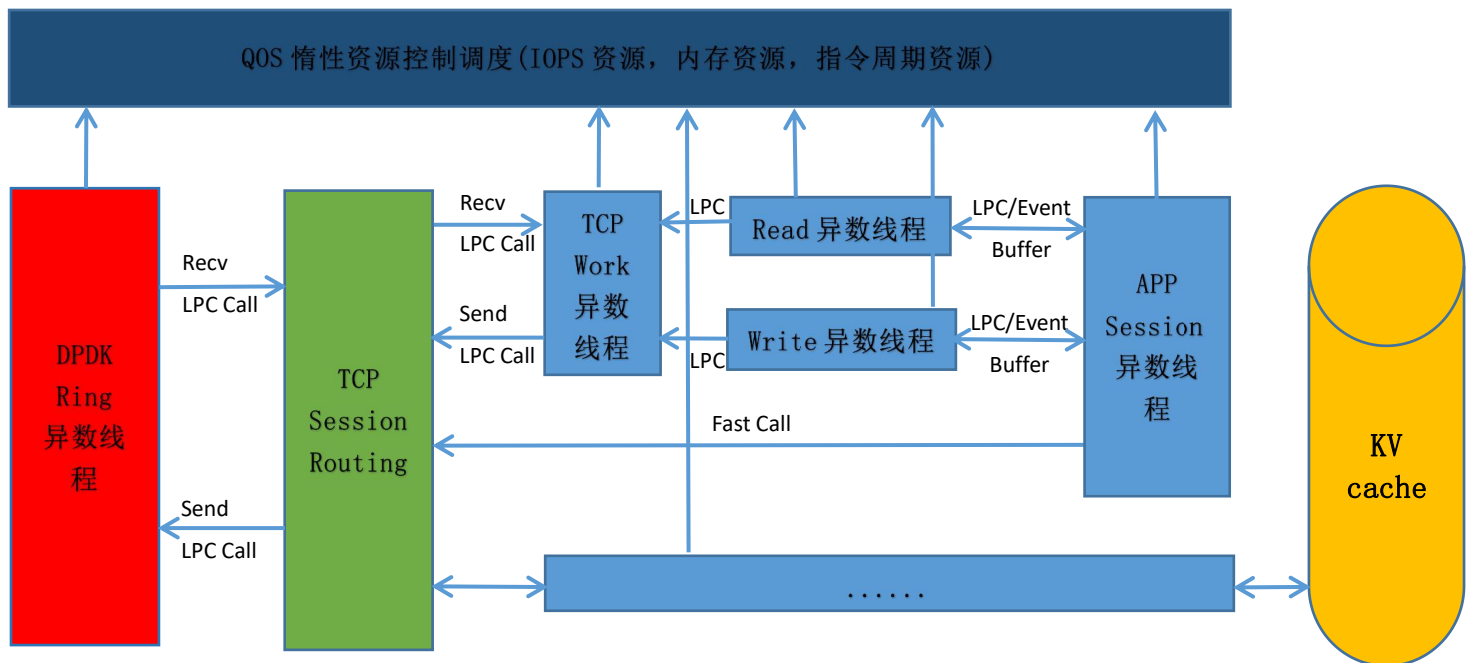
异数操作系统特性：

2. 同步阻塞线程模型，相对异步 IO 模型，有效的简化了复杂逻辑的设计能力，提供了类似 actor CSP 等编程模型体验，配合 DDPK RSS 等技术，实现了 CPU 核本地化业务无锁（硬件锁），逻辑表达却是严格阻塞顺序的，并有效填充了 DDPK 低效轮询时的垃圾时间占用。

4. 使用巨页对象池线程池技术，对象以及线程新建销毁等都是 0 开销，相比带 gc 的系统，系统性能更优，不会出现 gc 带来的系统雪崩体验。

1. 海量线程，高性能 Event 生成能力，线程切换能力，每线程 40 字节，轻载环境，每 CPU 核 5000W 每秒线程切换，3000W Event 生成完成能力，1000W 定时器能力，而依赖 epoll posix 阻塞 linux 线程的 mtcp ans f-stack 在 IO 队列复用 Event 的情况下，仅有 20-40W 的 Event (IO) 生成完成能力，仅仅达到 seastar 能力的十分之一，异数 OS 在 64 位环境中 1 亿线程仅 4G 内存，相对比 golang 的 2K 字节协程，在 IO 密集应用环境中（高 cache miss），异数操作系统在系统容量和性能上都是 go 程的 50 倍。

3. 惰性资源 QOS 技术，高效调度，降低资源需求的关键，惰性资源理论建立在海量同步阻塞线程技术基础上，其在目标资源不足时，将请求的线程挂起，在目标资源（QOS）可用时将线程唤醒继续，这样就减少杜绝了系统错误，解决了雪崩问题，并且在多数时候，资源并不实际分配，当最终在设备层发生数据交换时才发生最终的内存拷贝，所以未来可以很自然的适配 RDMA 技术，惰性资源技术还降低系统对高性能定时器的需求，比如 DDPK-ring 无缓存可用，需要等待一个内存存储周期（30-200 个指令周期）后，可用，则 DDPK-ring 线程被挂起，其他资源可用的线程将被唤醒，在一个或若干个线程切换周期后，被 DDPK-ring 惰性挂起的线程则被恢复并拥有可用的 ring。



异数 TCP 协议栈特性：

2. 零中间缓存，在惰性资源 QOS 技术加持下，TCP 协议栈不再为应用开辟数据缓存，而是直接在 KV-cache 和 DDPK-ring 之间做数据拷贝甚至发生业务逻辑，数据设置。更细致的 QOS，包括内存资源，IOPS 资源，CPU 周期，甚至无线网络信号环境，网卡状态等，使得 DDOS 或者环境不可用带来的雪崩情况被严格控制。比如 DDOS 时 DDPK ring 线程将停止调度，用惰性资源 QOS 代替 TCP 拥塞控制算法，网卡丢包时，全局 IOPS 将被调节，TCP Cwnd 将被调低。

3. IOT IM 领域优化的 TCP 协议，解决 C10K 魔咒，不在做没有任何意义的拥塞环境带宽探测与竞争，而是使用 QOS 技术按照系统设计需求硬件环境，以及业务性能需求做 IOPS 调节，减小窗口尺寸至 MSS 大小，在配合惰性资源技术，使得每 TCP 会话下降到 300 字节，如果用 golang 按照上图的线程模型按照传统 TCP 协议栈的设计，预计内存需求至少会达到 12K 以上。

1. 高性能，每核 3M 的 TCP 会话新建能力（带业务线程）。海量长链接下，20M 的 echo 能力，10M kv-cache 查询能力，解决 C10K 问题，实做 1 亿链接（IOT IM 场景模式），内存占用仅为 30G，每会话 300 字节。

众筹合作方式：

1. 多合伙人基金，技术维护年金。
 2. 买断。
 3. 公益捐助，精神捐助，开源。
- 联系方式：QQ 1055783080 电话：15026893648

