



岁寒

(<https://lvwenhan.com/>) 任何事情，从现在开始做，都不晚！

性能之殇（六）-- 现代计算机最亲密的伙伴：局部性与乐观

2018-11-19 / 阅读数：7704 / 分类：操作系统 (<https://lvwenhan.com/sort/36>)

冯·诺依曼架构中，指令和数据均存储在内存中，彻底打开了计算机“通用”的大门。这个结构中，“线性数组”内存天生携带了一个涡轮：局部性。

局部性分类

空间局部性

空间局部性是最容易理解的局部性：如果一段内存被使用，那么之后，离他最近的内存也最容易被使用，无论是数据还是指令都是这样。举一个浅显易懂的例子：

循环处理一个 Array，当处理完了 [2] 之后，下一个访问的就是 [3]，他们在内存里是相邻的。

时间局部性

如果一个变量所在的内存被访问过，那么接下来这一段内存很可能被再次访问，例子也非常简单：

```
$a = [];  
if ( !$b ) {  
    $a[] = $b;  
}
```

在一个 function 内，一个内存地址很可能被访问、修改多次。

乐观

“乐观”作为一种思考问题的方式广泛存在于计算机中，从硬件设计、内存管理、应用软件到数据库均广泛运用了这种思考方式，并给我们带来了十分可观的性能收益。

乐观的 CPU

第一篇文章中的 L1 L2 L3 三级缓存和第二篇文章中的分支预测与流水线，均是乐观思想的代表。

乐观的虚拟内存

虚拟内存依据计算机内存的局部性，将磁盘作为内存的本体，将内存作为磁盘的缓存，用很小的性能代价带来了数十倍并发进程数，是乐观思想的集大成者。

乐观的缓存

Java 经典面试题 LRU 缓存实现，也是乐观思想的一种表达。

同样，鸟哥的 yac (<http://www.laruence.com/2013/03/18/2846.html>) 也是这一思想的强烈体现。

设计 Yac 的经验假设

1. 对于一个应用来说，同名的Cache键，对应的Value，大小几乎相当。
2. 不同的键名的个数是有限的。
3. Cache的读的次数，远远大于写的次数。
4. Cache不是数据库，即使Cache失效也不会带来致命错误。

Yac 的限制

1. key的长度最大不能超过48个字符. (我想这个应该是能满足大家的需求的, 如果你非要用长Key, 可以MD5以后再存)
2. Value的最大长度不能超过64M, 压缩后的长度不能超过1M.
3. 当内存不够的时候, Yac会有比较明显的踢出率, 所以如果要使用Yac, 那么尽量多给点内存吧.

乐观锁

乐观锁在并发控制和数据库设计里都拥有重要地位, 其本质就是在特定的需求下, 假定不会冲突, 冲突之后再浪费较长时间处理, 比直接每次请求都浪费较短时间检测, 总体的性能高。乐观锁在算法领域有着非常丰富而成熟的应用。

乐观的分布式计算

分布式计算的核心思想就是乐观, 由 95% 可靠的 PC 机组成的分布式系统, 起可靠性也不会达到 99.99%, 但是绝大多数场景下, 99% 的可靠性就够了, 毕竟拿 PC 机做分布式比小型机便宜得多嘛。下一篇文章我会详细介绍分布式计算的性能之殇, 此处不再赘述。

乐观的代价

出来混, 早晚是要还的。

乐观给了我们很多的好处, 总结起来就是一句话: 以微小的性能损失换来大幅的性能提升。但是, 人在河边走, 哪有不湿鞋。每一个 2015 年 6 月入 A 股的散户, 都觉得大盘还能再翻一番, 岂不知一周之后, 就是股灾了。

乐观的代价来自于“微小的性能损失”, 就跟房贷市场中“微小的风险”一样, 当大环境小幅波动的时候, 他确实能承担压力, 稳住系统, 但是怕就怕突然雪崩:

1. 虚拟内存中的内存的局部性突然大幅失效, 磁盘读写速度成了内存读写速度, 系统卡死
2. 分布式数据库的六台机器中的 master 挂了, 系统在一秒内选举出了新的 master, 你以为系统会稳定运行? master 挂掉的原因就是压力过大, 这样就会导致新的 master 瞬间又被打挂, 然后一台一台地继续, 服务彻底失效。例如: 「故障说明」对六月六日 LeanCloud 多项服务发生中断的说明 (<https://forum.leancloud.cn/t/leancloud/914>)

WRITTEN BY



JohnLui (<https://github.com/johnlui>)

程序员, Swift Contributor

相关日志:

性能之殇（四）-- Unix 进程模型的局限 (<https://lvwenhan.com/操作系统/495.html>)

性能之殇（七）-- 分布式计算、超级计算机与神经网络共同的瓶颈 (<https://lvwenhan.com/操作系统/498.html>)

软件工程师需要了解的网络知识：从铜线到HTTP（五）—— HTTP 和 HTTPS (<https://lvwenhan.com/操作系统/489.html>)

软件工程师需要了解的网络知识：从铜线到HTTP（四）—— TCP 和路由器 (<https://lvwenhan.com/操作系统/488.html>)

软件工程师需要了解的网络知识：从铜线到HTTP（三）—— TCP/IP (<https://lvwenhan.com/操作系统/487.html>)

标签: 性能之殇

(<https://lvwenhan.com/tag/%E6%80%A7%E8%83%BD%E4%B9%8B%E6%AE%87>)

性能 (<https://lvwenhan.com/tag/%E6%80%A7%E8%83%BD>) 局部性

(<https://lvwenhan.com/tag/%E5%B1%80%E9%83%A8%E6%80%A7>) 乐观

(<https://lvwenhan.com/tag/%E4%B9%90%E8%A7%82>) 虚拟内存

(<https://lvwenhan.com/tag/%E8%99%9A%E6%8B%9F%E5%86%85%E5%AD%98>)

← 性能之殇（七）-- 分布式计算、超级计算机与神经网络共同的瓶颈 (<https://lvwenhan.com/操作系统/498.html>)

性能之殇（五）-- DPDK、SDN 与大页内存 →
(<https://lvwenhan.com/操作系统/496.html>)

发表评论：

昵称

邮件地址 (选填)

个人主页 (选填)

发表评论

友情链接： #Mukti's Blog (<http://www.feizhaojun.com/>) #住范儿 (<http://www.zhufaner.com/>) #Arron.y
(<http://blog.helloarron.com/>) 京ICP备13030650号-2 (<https://beian.miit.gov.cn>)

© 2011-2022 岁寒 (<https://lvwenhan.com/>) | Powered by Emlog (<http://www.emlog.net/>)