程序员练级攻略（2018）：异步I/O模型和Lock-Free编程

2018-06-28 陈皓，杨爽

程序员练级攻略（2018）：异步I/O模型和Lock-Free编程

陈皓，杨爽

00:53 / 13:40

异步I/O模型

异步I/O模型是我个人觉得所有程序员都必需要学习的一门技术或是编程方法，这其中的设计模式或是解决方法可以借鉴到分布式架构上来。再说一遍，学习这些模型，是非常非常重要的，你千万要认真学习。

史蒂文斯（Stevens）在《UNIX网络编程》一书6.2 I/O Models中介绍了五种I/O模型。

- 阻塞I/O
- 非阻塞I/O
- I/O的多路复用（select和poll）
- 信号驱动的I/O（SIGIO）
- 异步I/O（POSIX的aio_functions）

然后，在前面我们也阅读过了 - C10K Problem 。相信你对I/O模型也有了一定的了解。 这里，我们需要更为深入地学习I/O模型，尤其是其中的异步I/O模型。

首先，我们看一篇和Java相关的I/O模型的文章来复习一下之前的内容。Thousands of Threads and Blocking I/O: The Old Way to Write Java Servers Is New Again (and Way Better) ，这个PPT中不仅回顾和比较了各种I/O模型，而且还有各种比较细节的方案和说明，是一篇非常不错的文章。

然后，你可以看一篇Java相关的PPT - 道格·莱亚（Doug Lea）的 Scalable IO in Java，这样你会对一些概念有个了解。

接下来，我们需要了解一下各种异步I/O的实现和设计方式。

- IBM - Boost application performance using asynchronous I/O ，这是一篇关于AIO的文章。
- Lazy Asynchronous I/O For Event-Driven Servers ，这篇文章也很不错。
- 另外，异步I/O模型中的 Windows I/O Completion Ports ,你也需要了解一下。如果MSDN上的这个手册不容易读，你可以看看这篇文章 Inside I/O Completion Ports。另外，关于Windows，Windows Internals 这本书你可以仔细读一下，非常不错的。其中有一节I/O Processing也是很不错的，这里我给一个网上免费的链接I/O Processing 你可以看看Windows是怎么玩的。
- 接下来是Libevent。你可以看一下其主要维护人员尼克·马修森（Nick Mathewson）写的 libevent 2.0 book。还有一本国人写的电子书 《Libevent深入浅出》。
- 再接下来是 Libuv。你可以看一下其官网的 Libuv Design Overview 了解一下。

我简单总结一下，基本上来说，异步I/O模型的发展技术是： select -> poll -> epoll -> aio -> libevent -> libuv。Unix/Linux用了好几十年走过这些技术的变迁，然而，都不如Windows I/O Completion Port 设计得好（免责声明：这个观点纯属个人观点。相信你仔细研究这些I/O模型后，你会得到你自己的判断）。

看过这些各种异步I/O模式的实现以后，相信你会看到一个编程模式——Reactor模式。下面是这个模式的相关文章（读这三篇就够了）。

- Understanding Reactor Pattern: Thread-Based and Event-Driven
- Reactor Pattern
- The reactor pattern and non-blocking IO

然后是几篇有意思的延伸阅读文章。

- The Secret To 10 Million Concurrent Connections -The Kernel Is The Problem, Not The Solution - C10M问题来了.....
- 还有几篇可能有争议的文章，让你从不同的角度思考。
  - Select is fundamentally broken

- Epoll is fundamentally broken 1/2
- Epoll is fundamentally broken 2/2

Lock-Free编程相关

Lock-Free - 无锁技术越来越被开发人员重视，因为锁对于性能的影响实在是太大了，所以如果想开发出一个高性能的程序，你就非常有必要学习 Lock-Free的编程方式。

关于无锁的数据结构，有几篇教程你可以看一下。

- Dr.Dobb's: Lock-Free Data Structures
- Andrei Alexandrescu: Lock-Free Data Structures

然后强烈推荐一本免费的电子书：Is Parallel Programming Hard, And, If So, What Can You Do About It?，这是大牛 保罗·麦肯尼（Paul E. McKenney） 写的书。这本书堪称并行编程的经典书，必看。

此时，Wikipedia上有三个词条你要看一下，以此了解并发编程中的一些概念：Non-blocking algorithm 、Read-copy-update 和 Seqlock。

接下来，读一下以下两篇论文 。

- Implementing Lock-Free Queues， 这也是一篇很不错的论文，我把它介绍在了我的网站上 ，文章为"无锁队列的实现"。
- Simple, Fast, and Practical Non-Blocking and Blocking Concurrent Queue Algorithms ，这篇论文给出了一个无阻塞和阻塞的并发队列算法。

最后，有几个博客你要订阅一下。

- 1024cores - 德米特里·伐由科夫（Dmitry Vyukov）的和 lock-free 编程相关的网站。
- Paul E. McKenney - 保罗（Paul）的个人网站。
- Concurrency Freaks - 关于并发算法和相关模式的网站。
- Preshing on Programming - 加拿大程序员杰夫·普莱辛（Jeff Preshing）的技术博客，主要关注C++和Python两门编程语言。他用C++11实现了类的反射机制，用C++编写了3D小游戏Hop Out，还为该游戏编写了一个游戏引擎。他还讨论了很多C++的用法，比如C++14推荐的代码写法、新增的某些语言构造等，和Python很相似。阅读这个技术博客上的内容能够深深感受到博主对编程世界的崇敬和痴迷。
- Sutter's Mill - 赫布·萨特（Herb Sutter）是一位杰出的C++专家，曾担任ISO C++标准委员会秘书和召集人超过10年。他的博客有关于C++语言标准最新进展的信息，其中也有他的演讲视频。博客中还讨论了其他技术和C++的差异，如C#和JavaScript，它们的性能特点、怎样避免引入性能方面的缺陷等。
- Mechanical Sympathy - 博主是马丁·汤普森（Martin Thompson），他是一名英国的技术极客，探索现代硬件的功能，并提供开发、培训、性能调优和咨询服务。他的博客主题是Hardware and software working together in harmony，里面探讨了如何设计和编写软件使得它在硬件上能高性能地运行。非常值得一看。

接下来，是一些编程相关的一些C/C++的类库，这样你就不用从头再造轮子了（对于Java的，请参看JDK里的Concurrent开头的一系列的类）。

- Boost.Lockfree - Boost库中的无锁数据结构。
- ConcurrencyKit - 并发性编程的原语。
- Folly - Facebook的开源库（它对MPMC队列做了一个很好的实现）。
- Junction - C++中的并发数据结构。
- MPMCQueue - 一个用C++11编写的有边界的"多生产者-多消费者"无锁队列。
- SPSCQueue - 一个有边界的"单生产者-单消费者"的无等待、无锁的队列。
- Seqlock - 用C++实现的Seqlock。
- Userspace RCU - liburcu是一个用户空间的RCU（Read-copy-update，读-拷贝-更新）库。
- libcds - 一个并发数据结构的C++库。
- liblfds - 一个用C语言编写的可移植、无许可证、无锁的数据结构库。

其它

- 关于64位系统编程，只要去一个地方就行了： All about 64-bit programming in one place，这是一个关于64位编程相关的收集页面，其中包括相关的文章、28节课程，还有知识库和相关的blog。
- What Scalable Programs Need from Transactional Memory ，事务性内存（TM）一直是许多研究的重点，它在诸如IBM Blue Gene/Q和Intel Haswell等处理器中得到了支持。许多研究都使用STAMP基准测试套件来评估其设计。然而，我们所知的所有TM系统上的STAMP基准测试所获得的加速比较有限。

  例如，在IBM Blue Gene/Q上有64个线程，我们观察到使用Blue Gene/Q硬件事务内存（HTM）的中值加速比为1.4倍，使用软件事务内存（STM）的中值加速比为4.1倍。什么限制了这些TM基准的性能？在本论文中，作者认为问题在于用于编写它们的编程模型和数据结构上，只要使用合适的模型和数据结构，程序的性能可以有10多倍的提升。
- Improving OpenSSL Performance ，这篇文章除了教你如何提高OpenSSL的执行性能，还讲了一些底层的性能调优知识。
- 关于压缩的内容。为了避免枯燥，主要推荐下面这两篇实践性很强的文章。
  - How eBay's Shopping Cart used compression techniques to solve network I/O bottlenecks ，这是一篇很好的文章，讲述了eBay是如何通过压缩数据来提高整体服务性能的，其中有几个比较好的压缩算法。除了可以让你学到相关的技术知识，还可以让你看到一种比较严谨的工程师文化。
  - Linkedin: Boosting Site Speed Using Brotli Compression ，Linkedin在2017年早些时候开始使用 Brotli 来替换 gzip，以此带来更快的访问，这篇文章讲述了什么是Brotli以及与其它压缩程序的比较和所带来的性能提升。
- 这里有两篇关于SSD硬盘性能测试的文章。Performance Testing with SSDs, Part 1 和 Performance Testing with SSDs Part 2 ，这两篇文章介绍了测试SSD硬盘性能以及相关的操作系统调优方法。
- Secure Programming HOWTO - Creating Secure Software ，这是一本电子书，其中有繁体中文的翻译，这本电子书讲了Linux/Unix下的一些安全编程方面的知识。

相关论文

- Hints for Computer System Design ，计算机设计的忠告，这是ACM图灵奖得主 Butler Lampson 在Xerox PARC工作时的一篇论文。这篇论文简明扼要地总结了他在做系

统设计时的一些想法，非常值得一读。（用他的话来说，“Studying the design and implementation of a number of computer has led to some general hints for system design. They are described here and illustrated by many examples, ranging from hardware such as the Alto and the Dorado to application programs such as Bravo and Star“。）

- The 5 minute rule for trading memory for disc accesses and the 5 byte rule for trading memory for CPU time，根据文章名称也可以看出，5分钟法则是用来衡量内存与磁盘的，而5字节法则则是在内存和CPU之间的权衡。这两个法则是Jim Gray和Franco Putzolu在1986年的文章。

  在该论文发表10年后的1997年，Jim Gray和Goetz Graefe 又在 The Five-Minute Rule Ten Years Later and Other Computer Storage Rules of Thumb 中对该法则进行了重新审视。2007年，也就是该论文发表20年后，这年的1月28日，Jim Gray驾驶一艘40英尺长的船从旧金山港出海，目的是航行到附近的费拉隆岛，在那里撒下母亲的骨灰。出海之后，他就同朋友和亲属失去了联系。为了纪念和向大师致敬，时隔10多年后的2009年Goetz Graefe又发表了 The Five-Minute Rule 20 Years Later (and How Falsh Memory Changes the Rules)。

  注明一下，Jim Gray，关系型、数据库领域大师。因在数据库和事务处理研究和实现方面的开创性贡献而获得1998年图灵奖。美国科学院、工程院两院院士，ACM和IEEE两会会士。他25岁成为加州大学伯克利分校计算机科学学院第一位博士。在IBM工作期间参与和主持了IMS、System R、SQL / DS、DB2等项目的开发。后任职于微软研究院，主要关注应用数据库技术来处理各学科的海量信息。

小结

好了，总结一下今天的内容。异步I/O模型是我个人觉得所有程序员都必需要学习的一门技术或是编程方法，这其中的设计模式或是解决方法可以借鉴到分布式架构上来。而且我认为，学习这些模型非常重要，你千万要认真学习。

接下来是Lock-Free方面的内容，由于锁对于性能的影响实在是太大了，所以它越来越被开发人员所重视。如果想开发出一个高性能的程序，你非常有必要学习 Lock-Free的编程方式。随后，我给出系统底层方面的其它一些重要知识，如64位编程、提高OpenSSL的执行性能、压缩、SSD硬盘性能测试等。最后介绍了几篇我认为对学习和巩固这些知识非常有帮助的论文，都很经典，推荐你务必看看。

下篇文章是数据库方面的内容，我们将探讨各种类型的数据库，非常有意思。敬请期待。

下面是《程序员练级攻略（2018）》系列文章的目录（持续更新中）。

江小田

2018-06-28

专栏本来就是方向指导性质的，并不会有什么可以让你直接看了就能用，能涨工资的所谓干货，订阅前都提醒过了，干嘛还有那么多抱怨？

yzz

2018-06-28

如果一本书一篇文章就能让你精通某个技术，那说明这个技术本身就没什么难度（价值），技术就是要静下心来。给这么多干货还不知道感激，感谢陈老师。

乔海軍

2018-06-28

老陈领进门，修行靠个人，感谢陈老师给出的这些资料。

作者回复

2018-06-28

不客气

李小红

2018-06-28

全部放链接？一篇文章下来全是推荐看其他文章，有时间看那么多文章还订阅这个嘛？

作者回复

2018-06-28

开篇语已说过，一是这系列的文章只是在画地图，别人写的知识点给系统性的串起来，二是这里没有速成，三，高手区里就是扩大知识面。

我只能给告诉你要吃什么，而不会喂到你嘴里，更不可能代替你去吃。见谅！

666

2018-06-28

看了很多文章下的评论，感觉挺奇怪的。不少人期待的是什么呢，看几篇文章就希望能成为大神？都总想着走捷径，学习是持续的过程，能力也不是一朝一夕就起来的，都只看到大神的风光，却没有看到别人背后的努力跟付出。

Phoenix

2018-06-28

每周二和周四最期待的事情就是看耗子叔的专栏更新啦

Bin

2018-06-29

补英语...

栗子。

2018-06-28

正在刷leetcode的数据库篇，耗子叔的专栏不止让我见识到了更广阔的世界，也让我明白了接下来的路该怎么走

李文

2018-06-28

这类文章很好，囊括相关技术的书籍、博客、论文和框架库，有视野之广度和深度，相比纯粹技术知识分享更有裨益。

yzz

2018-06-28

感谢陈老师

---

poetess

2018-06-28

对新手一点都不友好啊这个系列

作者回复

2018-06-28

你已进入高手篇了.....新手请去新手区��

gatspy

2018-07-13

惊叹于耗子哥这么丰富的积累，这个系列的推荐书籍和文章看着都把人吓坏了��。 八年的沉淀都在这里了吧！
顺便问下c10k在哪个系列读过了呢��

马广乐

2018-07-10

谢谢陈老师，让我们知道了这么多不知道的东西。

我是一个正常名字��

2018-07-06

有人问农夫："种子麦子了吗？"农夫："没，我担心天不下雨。"那人又问："那你种棉花没？"农夫："没，我担心虫子吃了棉花。"那人再问："那你种了什么？"农夫："什么也没种，我要确保安全。"
这个故事告诉我们顾虑太多，思虑太多，就会导致束手束脚，一事无成。老了，不如放开自己，大胆去尝试。

干脆面君

2018-07-04

很好

蒋宏伟

2018-07-02

为什么伟大的程序员，比如比尔盖茨、扎克伯格、Vitalik Buterin，开创了伟大的企业？这些天才程序员和普通程序员的差异，是上述练级知识差异造成的？还是另有其他原因？

---

renwotao

2018-07-01

从工作后一直关注陈老师的文章，谢谢你的指引。

啊小炜哥

2018-06-30

请问一下，哪里可以看到链接文章的网址，我想在笔记本上看这些文章。

怀中抱小妹妹

2018-06-30

感谢、如果是自己筛选是万万做不到的

---

zliweijk

2018-06-30

酷壳网那个"High 一下！"笑死我了，哈哈哈��

---

paul.yang

2018-06-29

太吊了。只是心里有点慌。一个非科班出身的，这个得学多久。站在山脚仰望山顶。

---

echo

2018-06-29

看了这个系列的文章，惊叹于是如何收集如此多的链接。想问下耗子哥，这些链接是平常查询时遇到，觉得很好就分类整理了，并经常查看（手册类）或偶尔回看（经验类）?还是先收藏下来，每一段时间有一个整理的习惯?

作者回复

2018-06-29

基本上都是我自己的学习的过程总结，加上一些资料更新（后面还更多）。另外，我已经裁剪掉了七成了.....

纵横四海1949

2018-06-29

绝对干货

---

yun

2018-06-29

能否写一篇关于排查线上问题的文章?

---

dancer

2018-06-29

台上一分钟，台下十年功!

浪子恒心

2018-06-28

等到了一定境界，才会体会到这些东西是多么有用，非常感谢这些整理和推荐。

浪子恒心

2018-06-28

估计得修行十年。

杨雪峰

2018-06-28

今年补了一下英语，虽然文章里面还是有很多单词不认识，但是现在已经乐于看英文文章了。

Geek_122dd9

2018-06-28

得备一本牛津词典在桌子旁边了(ˇ•ω•ˇ)

黑球球

2018-06-28

耗子牛逼啊，高手牛逼的地方也就在于此，能把复杂世界中的零散信息关联起来，并且按照自己的能力梯度逐步了解和掌握，最后为我所用

---

akaQin

2018-06-28

昨天刚巧在学习IO五大模型，今天的专栏就讲到了，赞��

夜行观星

2018-06-28

新手先看入门，专业基础，一步一步来

心易修心

2018-06-28

这是高手进阶篇，不是针对新手的

Geek_8c5341

2018-06-28

感觉要学的内容好多，全部是英文，挺考验人的