

关于RTFM,STFW和RTFSC的感悟和理解

在我参与的高性能计算实验课程中，DGEMM（Double-precision General Matrix Multiply）是我们学习的焦点。从最初的朴素实现开始，我们逐步深入探讨了如何优化这一任务。在这个过程中，RTFM、STFW和RTFSC成为了至关重要的指导原则，不仅帮助我们提高效率，还深刻地影响了我对学习以及未来工作遇到问题的处理方式。

首先是：

RTFM (Read The Fucking Manual) :

在实验初期，我们的任务是实现一个朴素的DGEMM，这需要理解线性代数的基本原理和矩阵相乘的算法。最初，我有一种自以为是的想法，认为自己可以凭记忆和直觉完成这个任务，而随着课程的深入，DGEMM的实现逐步优化也逐渐变得复杂。随之而来的是，我发现，实际情况比我这个Linux新手想象的要复杂得多。我开始仔细阅读线性代数库的文档，深入了解DGEMM函数的参数和最佳实践。我发现，文档中包含了丰富的信息，可以帮助我更好地理解和使用这一函数。逐渐，我明白了RTFM的真正含义：文档是解决问题的宝贵资源，应该是我们的首选参考。

接着是：

STFW (Search The Fucking Web) :

随着我们的课程进展，我们的任务不再局限于基本的DGEMM实现。我们开始探索如何通过引入开源数学库（如OpenBLAS）来提高性能。这需要了解库的特性、配置选项以及优化技巧。尽管课程资料提供了一些信息，但它们并不能覆盖所有方面，在遇到一些未涉及到的问题时，就需要去Google等搜索引擎去寻找解决这些问题的方式。这时，STFW成为了非常有力的工具。我常常在互联网上查找开源库的优化建议、使用案例和性能调整方法，尤其是如今ChatGPT的流行，我们有了更新颖的解决问题的渠道。通过STFW，我找到了来自世界各地的专家分享的宝贵资源，这些资源在提高DGEMM性能方面发挥了关键作用。STFW不仅拓展了我的知识，还培养了我主动寻找解决问题的能力。

最后是：

RTFSC (Read The Fucking Source Code) :

随着我们的研究深入，我们开始考虑如何优化DGEMM的多线程执行。这涉及到深入了解多线程编程和并行计算。为了更好地理解OpenBLAS库的多线程实现，我开始阅读源代码。这是一个具有挑战性的任务，因为源代码通常相当复杂。然而，通过RTFSC，我不仅学到了多线程编程的核心概念，还深入了解了库的内部工作原理。这对于我们在实验中实现单线程分块和多线程分块的DGEMM版本至关重要。通过RTFSC，我学到了如何理解和改进复杂系统的关键技能。

除了上述的实例，我们的课程还考虑了其他性能优化策略，如使用OpenMP和MPI。这些方法都需要深入学习和探索，而RTFM、STFW和RTFSC的思维方式贯穿始终，帮助我们充分利用文档、互联网资源和源代码，以不断提高DGEMM的性能。

总的来说，在这次学习Linux和计算性能优化的路上，我开始意识到RTFM、STFW和RTFSC不仅仅是技术口号，它们代表了一种主动学习和自我提高的态度。通过不断运用这些方法，不仅可以取得实验课程中的成功，还培养了我解决问题的能力 and 应对技术挑战的信心。这些经验将对我们的职业发展产生深远的影响，因为它们不仅适用于高性能计算，还适用于各种技术领域。通过RTFM、STFW和RTFSC，我们培养了不断学习和适应不断变化的技术环境的能力，这将使我们成为更出色的技术从业者，在这里也对老师从课程开始至今一直提供课程资料并潜移默化地让我们掌握RTFM、STFW和RTFSC这3条方法表示感谢。