

1、计算载体与计算系统的关系？说明为什么目前大语言模型(LLM)使用的是 GPU，不是 CPU？说明驾驭 LLM 辅助编程，类似于共生小组合作逻辑？

<计算载体与计算系统的关系>

- 1、计算系统包含计算载体。
- 2、计算系统还包括在计算载体上运行的程序、数据，以及各部分和部分之间组织起来的设计思想等。
- 3、计算载体只是承载思想和数据运行的实体而已。

人类的计算方法和工具自古以来不断的发展进步。由一开始的手工、机械计算，到现在的电子计算，而未来将会出现光、生物和量子计算，计算的载体不断发展变化着，一方面说明了人类科学技术的进步，认识和改造世界的能力的加强。我们通过对客观世界的认识和改造，也在延伸着人本身的能力。另一方面则说明，物质世界的本质具有数学特性，具有一致性，不管工具如何发展，计算的本质并没有发生改变。人类的数学与计算就是一种对于宇宙本质规律的抽象与总结。因此无论载体是否变为光和生物、量子，计算速度无论如何提高计算的本质是不变的。

<为什么目前大语言模型(LLM)使用的是 GPU>

从对并行处理的需求，GPU 设计用于高度并行的处理任务，它包含成百上千的小处理核心(核)，这使得它能同时处理大量的操作。模型涉及到的矩阵乘法和向量操作可以在这些核心上高效地并行执行。对于数据吞吐率，GPU 具有更高的内存带宽，能够快速读写大量数据这对于训练和运行大语言模型尤为关键，因为这些模型需要处理和生成大量数据点。例如 NVIDIA 的 Tesla 和 Ampere 架构，包含专门的硬件单元(如 Tensor Cores)，用于加速深度学习的矩阵乘法和卷积运算。这些硬件加速器进一步提升了大语言模型的训练和推理速度。

从代价上说，尽管 GPU 的绝对功耗可能高于 CPU，但在每瓦特性能(即每单位能耗的处理能力)上，GPU 通常提供更好的性能，这使得使用 GPU 在大规模计算任务中更为节能。虽然单个 GPU 的成本可能高于 CPU，但考虑到其处理速度和并行能力，GPU 在处理深度学习模型时提供了更高的成本效益

<LLM 辅助编程的合作逻辑>

- 1、问题解析与任务理解：编程过程中，您可以向模型提出问题，请求解释特定概念、算法或语言特性的工作原理。模型可以提供解释、示例代码或者指向相关资源的链接，帮助您更好地理解问题和任务。
- 2、代码生成与示例展示：模型可以生成代码片段，从简单的语法结构到复杂的算法实

现，以满足您的需求。这些生成的代码片段可以作为起点，帮助您快速开始编程，并提供了一种学习新技术或语言特性的方式。

3、错误诊断与调试支持：在遇到错误或异常时，模型可以帮助您进行诊断，并提供可能的原因和解决方案。通过分析错误消息、代码逻辑和上下文信息，模型可以提供有针对性的建议，帮助您更快地解决问题。

4、算法设计与优化建议：在面临需要设计算法或者优化性能的任务时，模型可以提供各种算法设计思路、优化技巧和最佳实践。这些建议可以帮助您更高效地解决问题，提高代码质量和性能。

5、库和框架选择：在选择合适的库、框架或工具时，模型可以提供基于需求和条件的建议。根据您的项目要求和约束条件，模型可以推荐适用的工具，帮助您做出明智的选择。

6、学习资源推荐：模型可以推荐学习资源，包括教程、文档、论坛和社区，帮助您深入学习编程技能并解决具体问题。

7、版本控制和团队协作：模型可以提供版本控制工具的建议，帮助您管理代码库和团队协作。它可以推荐使用的版本控制系统、最佳实践和 workflows，以确保项目的顺利进行和高效交付。

2、借助下图中不同类型硬件的能耗级别提示，综述说明 INTEL 与 ARM 架构的不同能耗观？ 生物、量子计算大量降低能耗吗？

Intel 处理器架构的能耗观是以现有工艺资源追求最大通用计算能力为目标的，ARM 则是以完成一定的计算能力，实现能耗最小为目标的。生物、量子计算可以大量降低能耗，生物计算的能量消耗仅相当于普通计算机的十亿分之一。

3、怎样理解云端组织型嵌入式系统的编程特征？依据下图提示，说出并解释苹果 App 应该审核中 5 个主要方面要求，如何理解苹果云端类产品的封闭式开放生态特征？鸿蒙 OS 与其异同是什么？

<云端组织型嵌入式系统的编程特征>

云端系统可以简单的理解为云服务与终端计算并举形成的计算系统。

云计算操作系统，是云计算后台数据中心的整体管理运营系统（也有人认为云计算系统包括云终端操作系统，例如现在流行的各类手机操作系统，这与先行的单机操作系统区别不大，在此不做讨论），它是指构架于服务器、存储、网络等基础硬件资源和单机操作系统、

中间件、数据库等基础软件管理海量的基础硬件、软件资源之上的云平台综合管理系统。

<苹果 App 审核中 5 个主要方面要求>

- (1) 一致性原则。
- (2) 准确性原则。
- (3) 布局合理化原则。
- (4) 操作系统合理性原则。
- (5) 系统响应时间原则。

<苹果云端类产品的封闭式开放生态特征>

苹果的 iPhone 和 iPad 产品都是典型的云端组织型嵌入式产品，其封闭特征是指全球范围内只有一个云支持服务系统（iTunes+AppStore），开放是指通过其开发工具平台，及一系列的指南、审查机构，可以有众多的开发机构、开发者在其平台上进行大量的有商业价值的应用开发。

<鸿蒙 OS 与其异同是什么>

1、开放性和封闭性：

苹果云端产品： 苹果的云端产品（如 iPhone 和 iPad）在生态系统上具有封闭性。它们仅支持通过苹果的云服务系统（iTunes+App Store）提供的渠道获取应用。开发者需要遵守苹果的开发工具平台和审查机构的规定。

鸿蒙 OS： 鸿蒙 OS 在设计上更加开放，允许开发者使用其开放的开发工具和平台来构建应用程序。这使得开发者可以更自由地创建和分发应用，而不受限于单一的应用商店。

2、开发者生态系统：

苹果云端产品： 苹果的生态系统相对封闭，开发者必须遵守苹果的规定和审查程序才能在其平台上发布应用。

鸿蒙 OS： 鸿蒙 OS 的生态系统还在发展阶段，但从其开放性设计来看，它可能更容易吸引更多的开发者加入，并在该平台上构建应用程序。

3、应用分发方式：

苹果云端产品： 应用程序只能通过苹果的 App Store 进行分发和获取，这种方式保证了应用的安全性和一致性。

鸿蒙 OS： 由于鸿蒙 OS 更加开放，开发者可能会有更多的选择，可以通过不同的渠道来分发应用程序，而不仅仅局限于单一的应用商店。

4、审核和安全性：

苹果云端产品： 苹果通过严格的应用审核流程确保了应用的质量和安全性。

鸿蒙 OS： 鸿蒙 OS 的安全性和审核流程可能会受到不同的管理和实施方式的影响，这取决于华为在该方面的决策和执行力度。

5、用户体验和一致性：

苹果云端产品： 苹果注重用户体验和界面一致性，对应用的布局、操作系统的合理性和系统响应时间等方面有严格的要求。

鸿蒙 OS： 鸿蒙 OS 是否能提供与苹果产品一致水平的用户体验尚不明确，但其也可能注重这些方面，以确保用户对其产品的满意度。

4、通过本课程学习及相关内容阅读，借助下面示意公式的提示，说明自由或开源软件中的利他编程，极其基本质量控制？结合其中的“实用许可”，举 2 个主流软件许可证。

<利他编程及其基本质量控制>

利他编程是一个高层阶的大概念，讲到了编程的目的、价值归宿取向问题，但凡以利他编程驱动的事务，要有更为强烈的行业公共基础价值，其编程测试通常是自发的一体化的，专业化的，而且在合适的时候可以相对频繁的使用重写方式进行，这在利己编程中通常是不可思议的。基本质量可理解为基础质量包括代码组织、版本控制和系统应用的可靠、性能、架构兼容，以及 UI 等方面。

<主流软件许可证>

MIT 许可证、Apache 许可证