文本, 信件

AI 生成的内容可能不正确。

单指令多数据（SIMD）

向量化是把多个操作组合起来，实现并行执行，以此提高运算效率。

向量（SIMD）通道-通过向量寄存器上的向量操作的通道，用于单个数据元素，非常类似于多车道高速公路上的车道。

向量宽度-向量单元的宽度，通常以位表示。

向量长度-向量在一次操作中可以处理的数据元素的数量。

向量（SIMD）指令集-扩展常规标量处理器指令以利用向量处理器的指令集。

向量化主要适用于数据并行的场景，即对多个独立的数据元素执行相同的操作。当应用程序受限于内存带宽（大多数应用程序属于这种情况）时，单纯增加向量操作的 FLOPs（每秒浮点运算次数）可能无法带来显著的性能提升。因为此时性能瓶颈在于内存的读写速度，而不是 CPU 的运算能力。

生成指令-向量指令必须由编译器生成或通过内部函数或汇编程序编码手动指定

文本

AI 生成的内容可能不正确。

在C++中使用\_\_restrict或\_\_restrict\_\_属性。

文本

AI 生成的内容可能不正确。

OpenMP是线程和共享内存并行编程最广泛支持的开放标准之一。

每个线程（用曲线表示）都有自己的指令指针、堆栈指针和堆栈内存，但共享堆和静态内存数据。

文本, 信件

AI 生成的内容可能不正确。

OpenMP（Open Multi - Processing）是一种用于共享内存并行编程的应用程序编程接口（API），它的确主要通过线程来实现任务的并行化。其核心是 “fork - join” 并行执行模型：程序开始时以单线程（主线程）运行，当遇到并行区域（由 #pragma omp parallel 等编译指示定义），主线程会派生多个子线程，这些子线程并行执行并行区域内的代码；当并行区域执行完毕，子线程汇合到主线程，程序继续单线程执行后续代码。

文本, 信件

AI 生成的内容可能不正确。.MPI：并行主干

消息传递接口（MPI）标准的重要性在于，它允许程序访问更多的计算节点，从而通过向模拟中添加更多的节点来运行越来越大的问题。名称“消息传递”是指轻松地将消息从一个进程发送到另一个进程的能力。

MPI 程序总是在开始时调用 MPI\_Init，在程序结束时调用 MPI\_Finalize

消息本身在发送端和接收端总是由三元组构成：一个指向内存缓冲区的指针、一个计数和一个类型。发送的类型和接收的类型以及计数可以不同。

消息信封（由排名、标签、通信组构成的三元组）主要用于确定消息的发送路径和区分不同的消息，起到了消息路由和标识的作用：