

计算机建模 2

刘 钦

南京大学软件学院

不同的软件框架，解
决问题是不一样的

第二步 - 建立计算机模型

Functional Programming Paradigm

- $\text{Sum}(n) = n + \text{Sum}(n-1)$
- $\text{Sum}(1) = 1$

第三步 - 制订解决方案

- (define (sum n)
 - (if (= n 1)
 - 1
 - (+ n (sum (- n 1)))))

第三步 - 制订解决方案

- (sum 6)
- (+ 6 (sum 5))
- (+ 6 (+ 5 (um 4)))
- (+ 6 (+ 5 (+ 4 (sum 3))))
- (+ 6 (+ 5 (+ 4 (+ 3 (sum 2)))))
- (+ 6 (+ 5 (+ 4 (+ 3 (+ 2 (sum 1)))))
- (+ 6 (+ 5 (+ 4 (+ 3 (+ 2 1)))))
- (+ 6 (+ 5 (+ 4 (+ 3 3))))
- (+ 6 (+ 5 (+ 4 6)))
- (+ 6 (+ 5 10))
- (+ 6 15)
- 21

不同的硬件框架，解
决问题是不一样的

SSE技术

- 1999年，Intel在其Pentium III微处理器中集成了SSE（Streaming SIMD Extensions）技术，有效增强了CPU浮点运算的能力。
- SSE兼容MMX指令，可以通过SIMD和单时钟周期并行处理多个浮点数据来有效提高浮点运算速度，对图像处理、浮点运算、3D运算、视频处理、音频处理等诸多多媒体应用起到全面强化作用。
- 具有SSE指令集支持的处理器有8个128位的寄存器，每一个寄存器可以存放4个单精度（32位）浮点数。SSE同时提供了一个指令集，其中的指令允许把浮点数加载到这些128位寄存器中，这些数就可以在这些寄存器中进行算术逻辑运算，然后把结果送回主存。也就是说，SSE中的所有计算都可以针对4个浮点数一次性完成，这种批处理带来了效率的提升。

解决这样一个问题

- 计算一个很长的浮点型数组中每一个元素的平方根

基于以前框架的解决方案

- for each f in array
- {
 - 把 f 从主存加载到浮点寄存器
 - 计算平方根
 - 再把计算结果从寄存器中取出写入主存
- }

基于SSE技术的解决方案

- for each 4 members in array //对数组中的每4个元素
- {
 - 把数组中的这4个数加载到一个128位的SSE寄存器中
 - 在一个CPU指令执行周期中完成计算这4个数的平方根的操作
 - 把所得的4个结果取出写入主存
- }