程序性能优化理论与方法

韩林　高伟　著

先进编译实验室
Advanced Compiler

目录

上篇

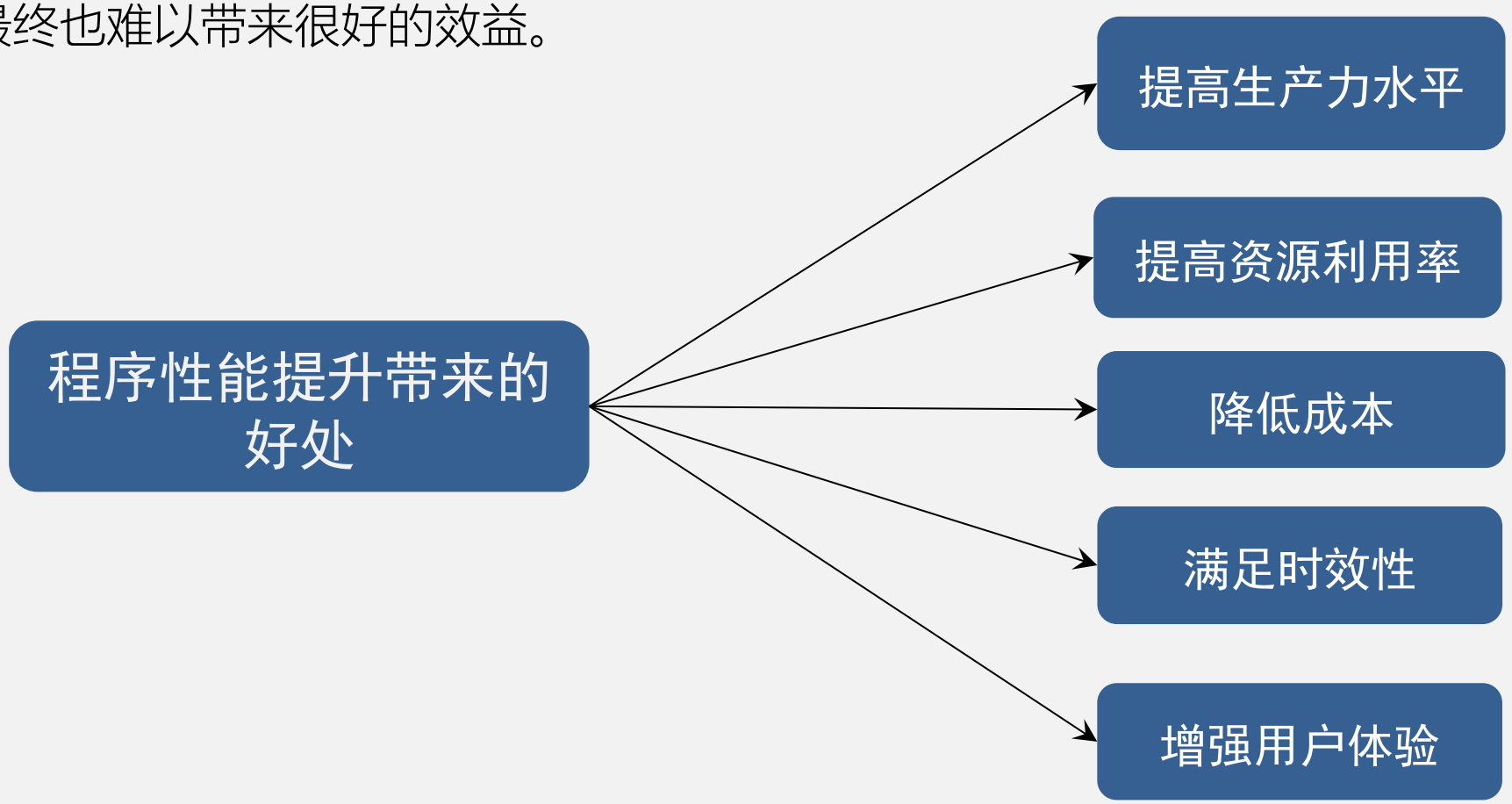
目录

下篇

程序性能优化的意义

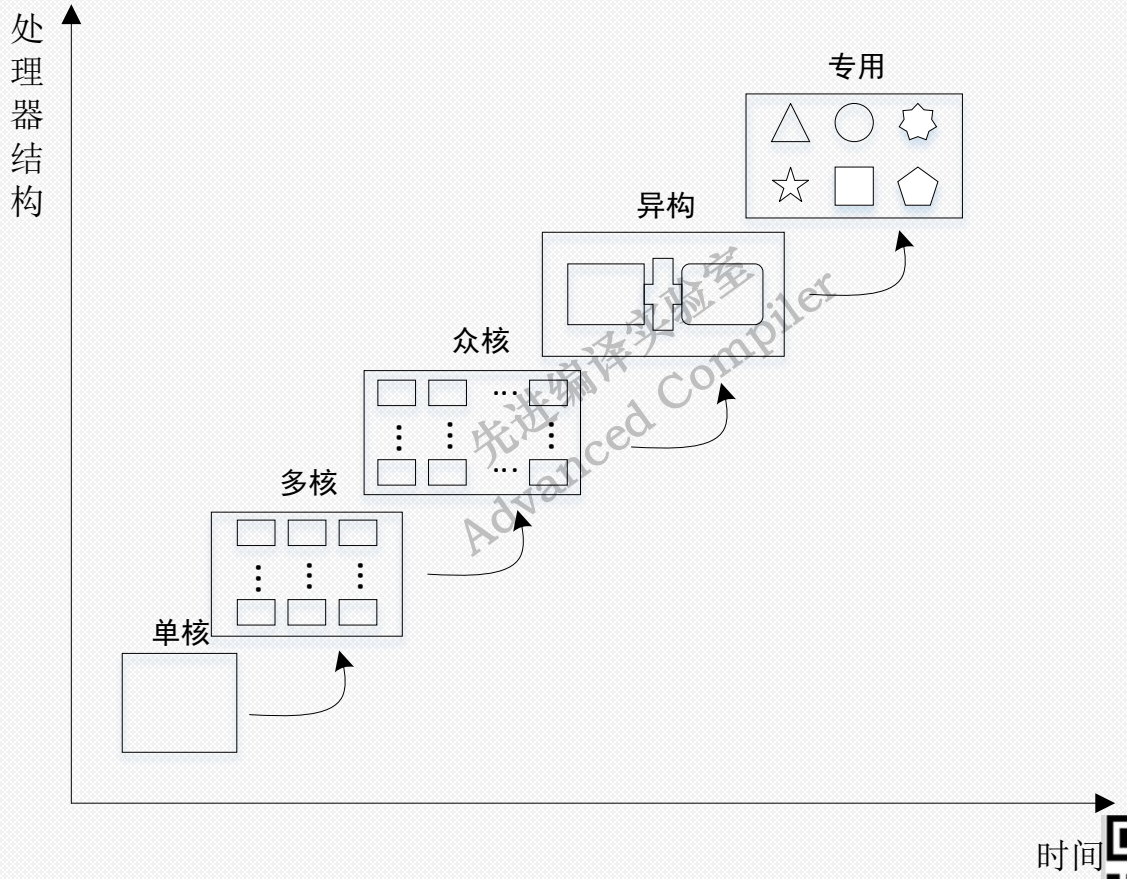
1.1 程序性能要求的不断提高

随着硬件计算能力的提高，对程序性能的要求也越来越高，复杂的需求很难完全依靠硬件解决。比如在开发软件时不重视性能优化，最终实现了功能要求，但运行效率低下，最终也难以带来很好的效益。



1.2 处理器架构的不断发展

纵观处理器的发展历程，其早期的性能增长基本遵从摩尔定律，但是随着主频的不断提升、芯片上集成的晶体管数的急剧增加，功耗、互连、复杂度也在指数级的增长，此时的摩尔定律不再适用于单核处理器。



1.2.1 单核结构

CPU时钟频率

超大规模集成电路工艺的发展，使得单颗芯片上可集成更多的资源，为处理器体系结构的发展提供了源源动力，通过改进处理器体系结构获得更高的时钟频率是单核处理器设计的重要方向之一。

执行效率

提高每个时钟周期内的执行效率的方式之一就是采用并行计算，分为指令级并行和数据级并行，指令级并行是指处理器同时执行多条指令，数据级并行是指对程序中的多个数据进行相同操作。

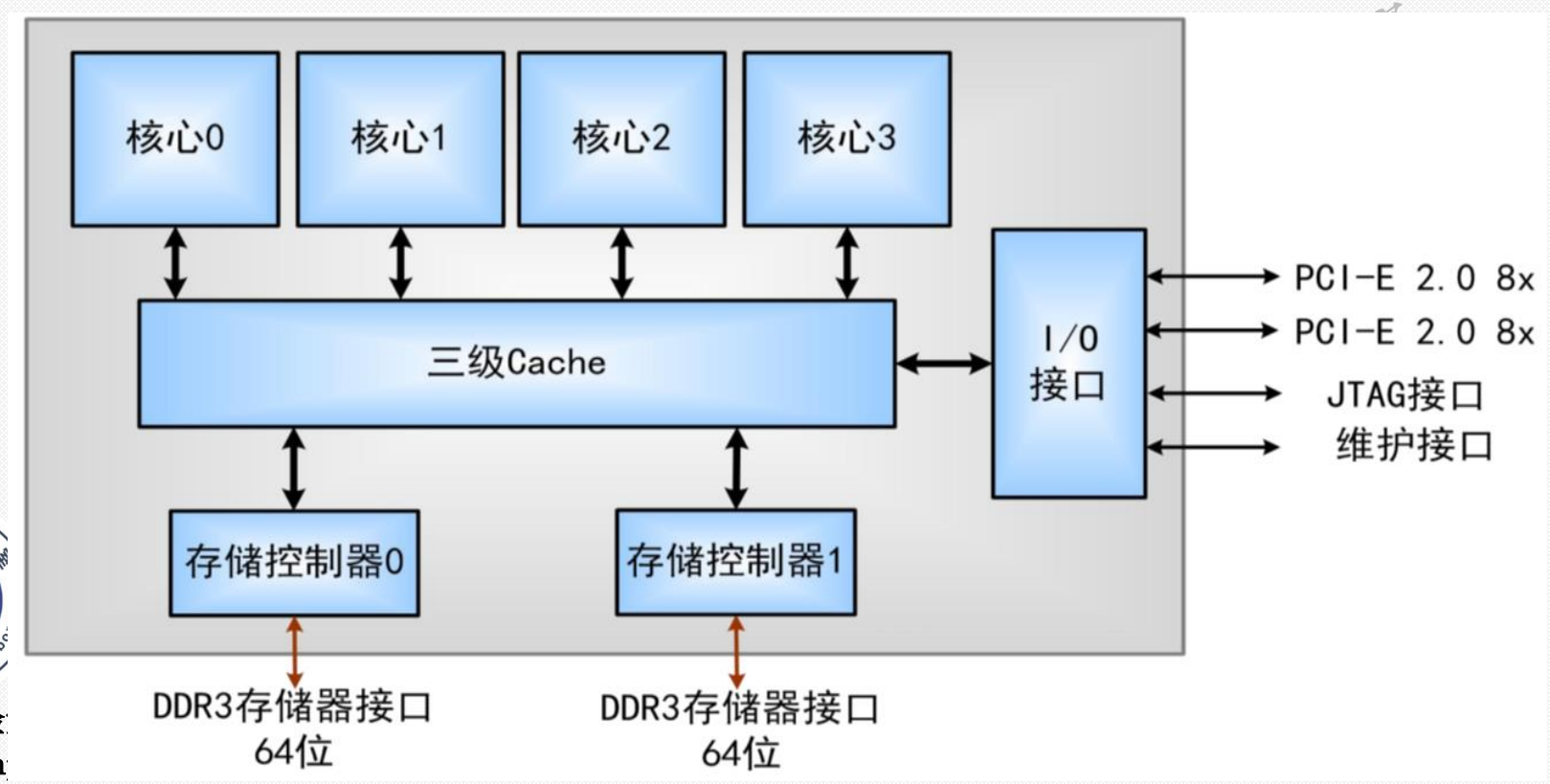
1.2.1 单核结构

表1　带有SIMD扩展部件的处理器

厂家	□理器	指令集	向量□度
Motorola	G4	AltiVec	128bits
DEC	Alpha	MVI	64 bits
SGI	MIPS V	MDMX	64 bits
Intel	Pentium	MMX	64 bits
		SSE	128 bits
	Core	AVX	256 bits
		IMCI	512 bits
Sony	Cell	AltiVec	128 bits
Sun	SPARC v9	VIS	64 bits
HP	PA-RISC	MAX-2	64 bits
AMD	Athlon	3DNow!	128 bits
ARM	ARMv6	NEON	128 bits
	PPC970	VMX	128 bits
IBM	P6	VMX	128 bits
	BG/L	/	256bits
□芯	Godson	/	256 bits

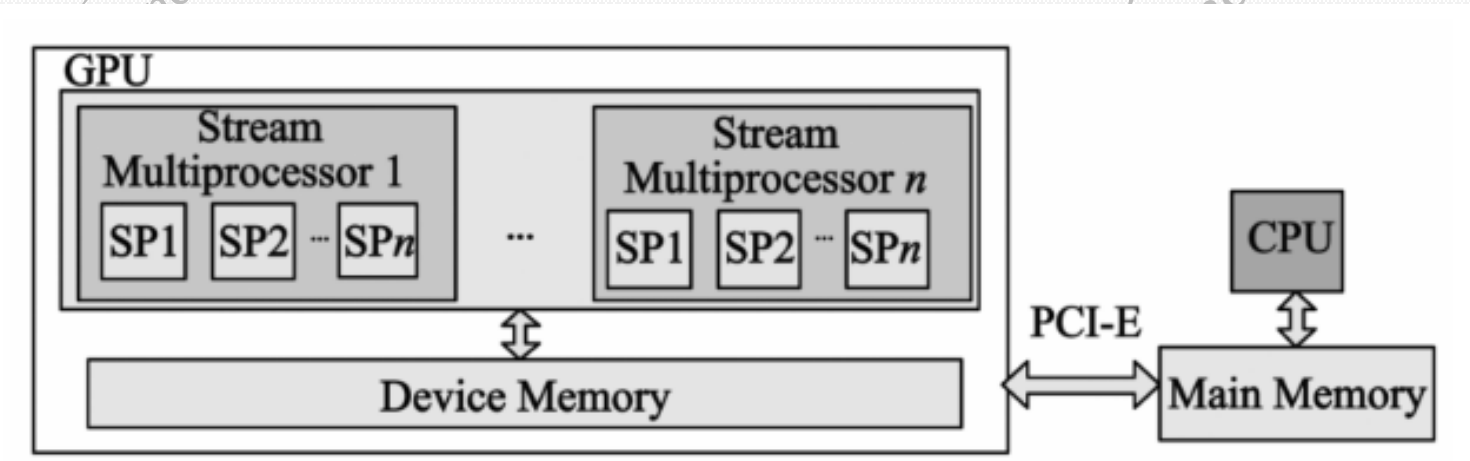
1.2.2 多核结构

多核处理器即在一个单芯片上集成多个处理器内核，其中每个核都是一个独立的物理处理器，多核处理器支持多个线程在多个处理器核上同时执行，使得整个处理器可同时执行的线程数目是单处理器的数倍，极大地提升了处理器的性能。下图为某多核处理器结构。



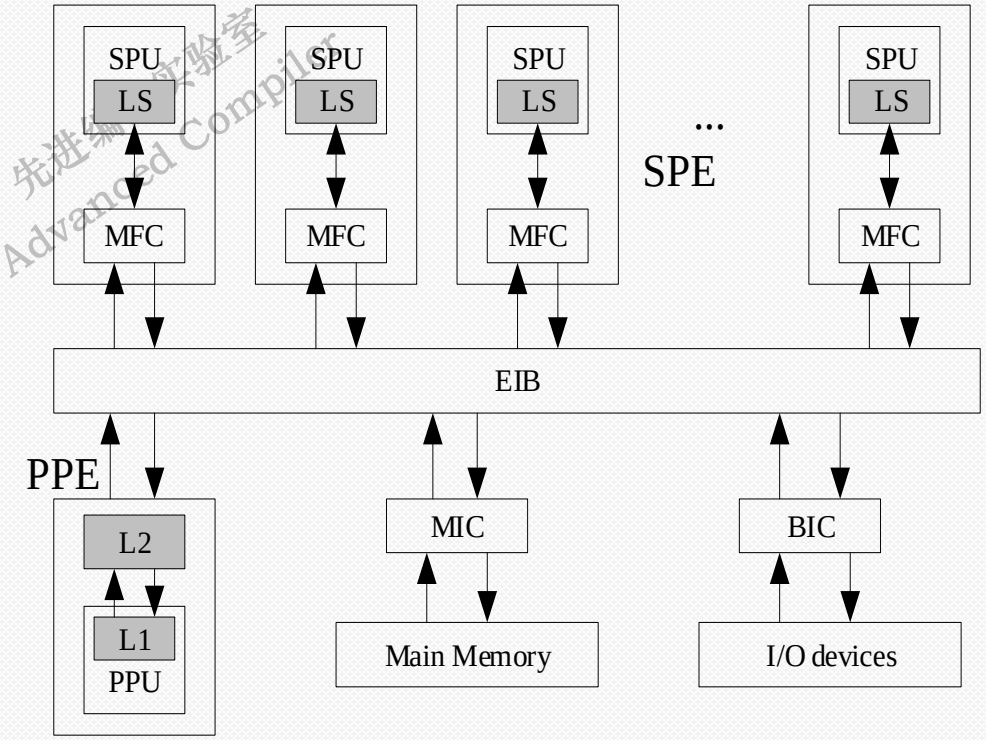
1.2.2 众核结构

众核处理器由几十到几千个功能较小的内核组成，使用针对更大并行性和吞吐量进行优化的嵌入式处理器，来解决数据瓶颈问题。其与多核之间的关键区别包括处理器核心的数量以及实际的处理器结构。最具代表性的众核处理器就是图形处理器（Graphics Processing Unit，GPU），下图为GPU系统架构。



1.2.4 异构结构

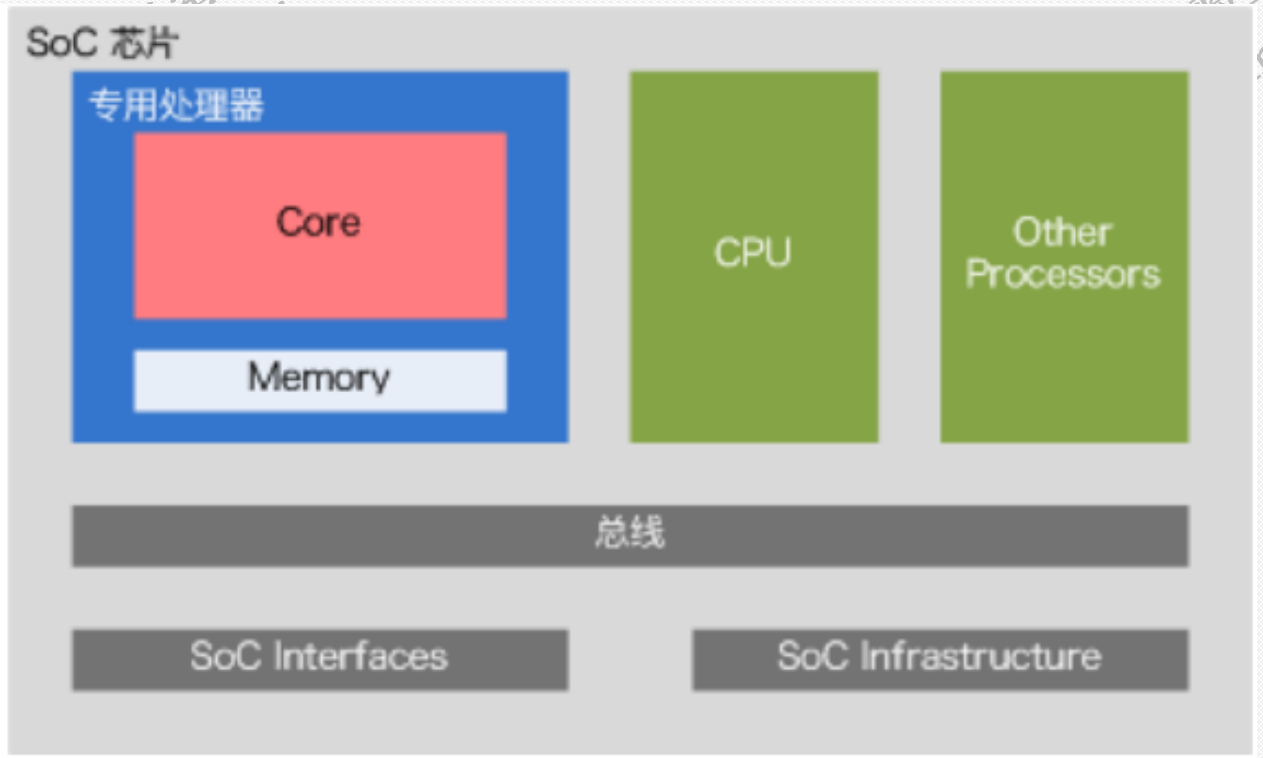
异构多核是指将功能或性能相异的处理器通过一定的互连结构连接起来，一般由通用处理器和专用加速处理器构成，在芯片内面向不同的指令集成了不同类型的计算部件，典型代表为索尼、IBM和东芝联合研发的Cell处理器。



MFC：内部存储器流控制器
MIC：内存控制器
BIC：外部IO接口
EIB：内部互联总线
LS：本地存储区

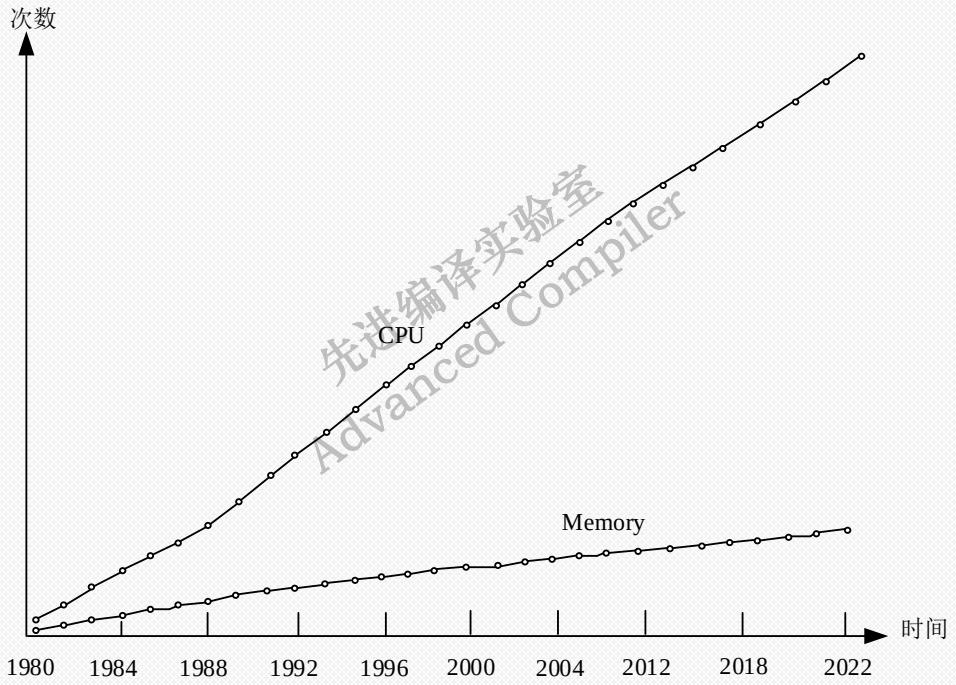
1.2.4 专用结构

专用处理器是通过将硬件架构进行定制并使其具备特定领域应用特征，使得该领域的一系列应用任务都能高效执行，例如在机器学习领域，比较有代表性的专用架构为Google的张量处理器TPU，专用于神经网络运算、神经机器翻译等诸多任务。



1.3 存储结构的不断发展

在过去的几十年中，处理器性能以每年50%至100%的速度平稳增长，而存储器的性能却只以每年7%左右的速度增长，可以断言的是未来处理器与存储器之间的速度差异将会越来越大，所以存储系统仍将是影响整个计算系统性能的一个关键瓶颈。处理器与存储器之间的速度差异如图所示。



1.3 存储结构的不断发展

现代处理器采用了多层次的、容量和性能不同的缓存，其中上一级缓存容量比下一级缓存小，但是延迟更小、带宽更大。如图展示了越靠近 CPU 的存储器速度越快，容量越小，每比特价格越高；而越远离CPU的存储器速度越慢，容量越大，每比特价格越低。如何让处理器的访问尽可能地发生在距处理器较近、访问时间较短的存储层次中是减少处理器访问延迟的关键。

1.3 存储结构的不断发展

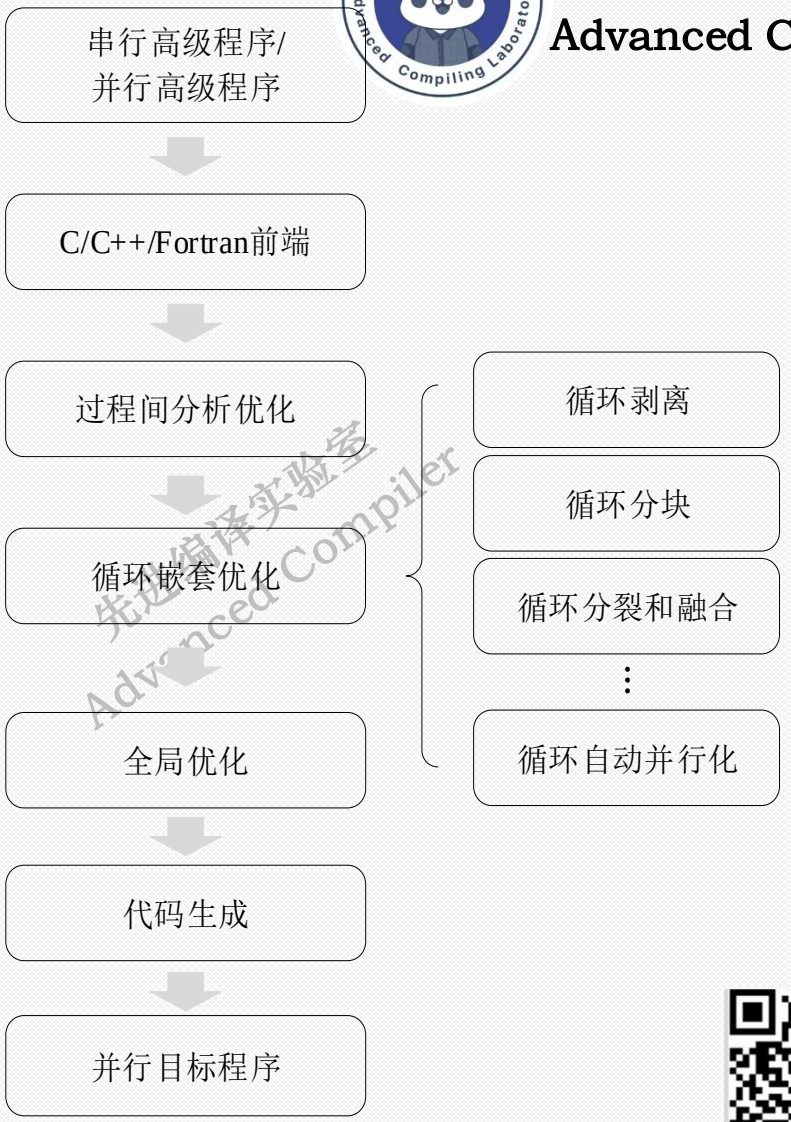
表2 距离处理器不同远近的存储层次的性能指标统计

存储层次/指标	寄存器	□存	主存	辅存
典型容量	<1KB	<804MB	<2TB	<6TB
访问时间（ns）	0.25-0.5	0.5-25	50-250	5, 000, 000
带宽（MB/s）	50, 000-500, 000	5, 000-20, 000	2, 500-10, 000	50-500
实现工艺	多端口定制CMOS	CMOS SRAM	CMOS DRAM	磁盘，软盘

1.4 编译器能力的局限性

编译器作为人机语言交互的桥梁，其功能是将程序员编写的高级语言程序翻译成面向目标机器的可执行程序。

并行化编译系统已成为高性能计算机系统中重要的组成部分，该编译系统通过自动检测程序中潜在并行性，将原始的串行语法成分自动地转变为等价的并行语句，极大地减少了程序员改写并行程序的工作量。



1.4 编译器能力的局限性

影响编译器效果的原因可概括为如下：

（1） 编译器采用的程序静态分析技术自身有能力的局限；

（2）编译策略的保守性原则制约了编译优化的能力；

（3）编译器要在通用性和高效性之间保持微妙的平衡；

（4）编译器采用何种优化策略还要考虑用户对编译时间的容忍度；

（5）体系结构与计算器件的快速升级与迭代对编译器充满挑战；

1.5 编程模型的局限性

并行编程模型通过抽象并行计算机体系结构，提供给程序员一种方便与算法结合的编程逻辑，常见的编程模型如CUDA、OpenACC、OpenCL、OpenMP、MPI等。但是当前的并行编程模型也有其局限性，如OpenMP主要针对于共享存储结构，MPI主要针对与分布存储结构。

为了减轻程序员的编码负担，编程模型倾向于隐藏底层设备的细节，但也意味着可能会限制了程序与底层硬件的深度结合，而使程序失去了获得更高性能的机会，表现在以下三个方面：

（1）编程模型无法充分表达程序的理想性能；

（2）编程模型无法充分发挥硬件的计算性能；

（3）编程模型对硬件支持的局限性也会引起程序性能的降低；

1.6 小结

本章阐述了处理器由单核到多核、众核、异构、专用结构的不断变化，以及适配的存储结构、编译器、编程模型的优缺点，引出程序存在着许多进一步性能提升的空间。本章中提出的程序在不同硬件运行平台的适配、在使用存储空间时需要合理的调度调试、使用编译器时需要了解其能力限制、利用编程模型时也会制约其性能等问题，直面解决这些问题是程序性能优化的主要内容。