高性能计算机体系结构模型

高性能计算机体系结构课程汇报

宋震，苏晓，郭冠男，张泰然，刘旭昭

2024-12-03

目录

1 概述

自弗林分类法 [7] 后，有关高性能计算机体系结构的讨论层出不穷。在弗林分类法中，计算机体系结构被分为四个大类：SISD(Single Instruction Stream, Single Data Stream)、SIMD(Single Instruction Stream, Multiple Data Stream)、MISD(Multiple Instruction Stream, Single Data Stream)、MIMD(Multiple Instruction Stream, Multiple Data Stream) 的。其中 SISD，即单指令流、单数据流计算机，常用于对计算效率要求不高的低端工业控制芯片，或仅以一个计算单元的身份作为组成高性能计算机的组成元件，因而无缘于本文的讨论。

而关于 MISD, 即单指令流多数据流计算机，目前主要存在两种不同的声音。第一种声音认为，多指令流单数据流仅在理论上可行而缺乏现实世界中的实例 [7]。另一种声音认为，诸如脉动阵列（Systolic Array）[12] 等矩阵运算的专用 AI 加速芯片，由于其中硬件编码实现了一些固定的指令序列，而数据从中并行得流入，因而在某种程度上被认为是 MISD 计算机的实例 [11]。但无论出于那种理由，我们都不难发现，MISD 难以实现一个易于编程的图灵完备系统，因此也无法承担起高性能计算机整体架构的重任，因而同样无缘于文本的讨论。

本文核心关注高性能计算机的整体架构的分类以及各分类间的联系，具体而言，本文将依照弗林分类法中的 SIMD 以及 MIMD 两种体系结构并对其展开讨论。为了对比不同体系结构，我们收集了 40 年以来在高性能计算机领域具有一定代表性的指令集、芯片或超算等共 15 个案例，试图从特殊到一般地地介绍这些不同的高性能计算机体系结构的共性与特性，进而力求给出我们自己对这些体系结构的一些对比、看法以及历史性的评述。

1.1 分类原则

高性能计算机先驱 Seymour Roger Cray[4] 曾经说：“If you were plowing a field, which would you rather use: two strong oxen or 1024 chickens?”（如果你要耕一块地，你愿意用 2 头牛还是 1024 只鸡呢?）。Cray 使用这句话反驳那些 MIMD 的拥护者，企图向他们论证自己的 SIMD 设计能更有效地解决高性能计算中的问题。

随着 MPP（Massively Parallel Processing）以及节点集群（Cluster）在高性能计算中应用的日渐普及，越来越多的人逐渐认可了 Cray 观点的对立面。尽管如此，SIMD 的高性能设计仍然并非一无是处，很多在个人计算机以及小型工作站上使用的商用 CPU（Central Processing Unit）中都采用了的向量指令集作为其指令集的拓展，其中也体现了一些 SIMD 的设计理念 [8]。至今更多研究者已经不再纠结于 SIMD 与 MIMD 的二择问题，而侧重关注于如何将 SIMD 与 MIMD 结合在一起以在“后摩尔定律时代”尽可能进一步提升计算机系统的性能 [9]。

因此，我们在本文中遵循了以下的分类原则，着重介绍了几类体系结构（详见图 1）。从指令流的角度我们首先将高性能计算机体系结构划分为了 SIMD 和 MIMD 两类，由于单机可拓展能力的限制，在 Top500 超级计算机列表中，目前绝大多数大多数超级计算机从顶层架构的角度来看均属于 MIMD 计算机 [26]。图 2 为 Top500 超级计算机中 MPP 与 Cluster 两类顶层架构的分布情况。图中绿色区域表示采用了 MPP 架构的超级计算机，蓝色的区域表示采用了 Cluster 架构的超级计算机，而每部分的面积正比于该计算机的浮点运算性能（单位 GFLOPS）。图中不难看出，在 Top500 列表中，大多数计算机采用了 Cluster 的架构，但其中性能最高的实例中仍然以 MPP 架构

为主。

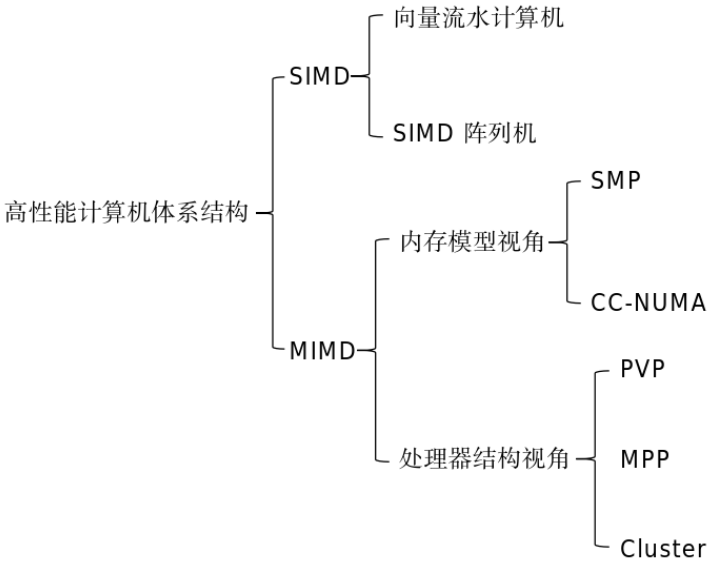


图 1: 高性能计算机体系结构分类思路

2 SIMD 架构结构介绍

2.1 SIMD 的概念

SIMD 是单指令多数据的缩写，它是一种并行计算技术，可以使得处理器在一个时钟周期内通过一条指令同时对多个数据进行操作。与传统的 SISD（单指令单数据）不同，SIMD 允许在同一指令下并行处理多个数据元素，极大地提高了计算效率，特别适用于大量相同运算的任务，如图像处理、科学计算、信号处理等。

与 SISD 相比，SIMD 架构因其高效性、能效和广泛的适用范围而备受青睐。通过并行化计算任务，SIMD 能够显著减少程序的执行时间，这使得它在处理大量数据时比传统的串行计算方式更加高效。此外，SIMD 架构在较低的能耗下能够处理更多的数据，展现出卓越的能效比。这种架构特别适用于多媒体应用、科学计算、机器学习中的矩阵运算等领域，因为它能够同时对多个数据执行相同的操作，从而加速这些领域的计算过程。

2.2 向量流水计算机

2.2.1 向量化与流水线化的概念

向量流水计算机是结合了向量处理和流水线技术的一种优化方案。向量化处理指的是对多个数据进行统一操作，在一条指令下同时处理多个数据元素（如向量加法、乘法等）。流水线技术将复杂的计算过程分成多个阶段，每个阶段可以同时处理不同的数据。在每个时钟周期内，不同的数据会在流水线的不同阶段被处理，最大化资源的使用。在这种架构下，SIMD 计算单元不仅并行处理多个数据元素，还通过流水线方式进行分阶段处理。这种设计可以充分利用硬件并行性，在避免了数据间依赖的前提下，可以大幅提升批量数据的处理速度。

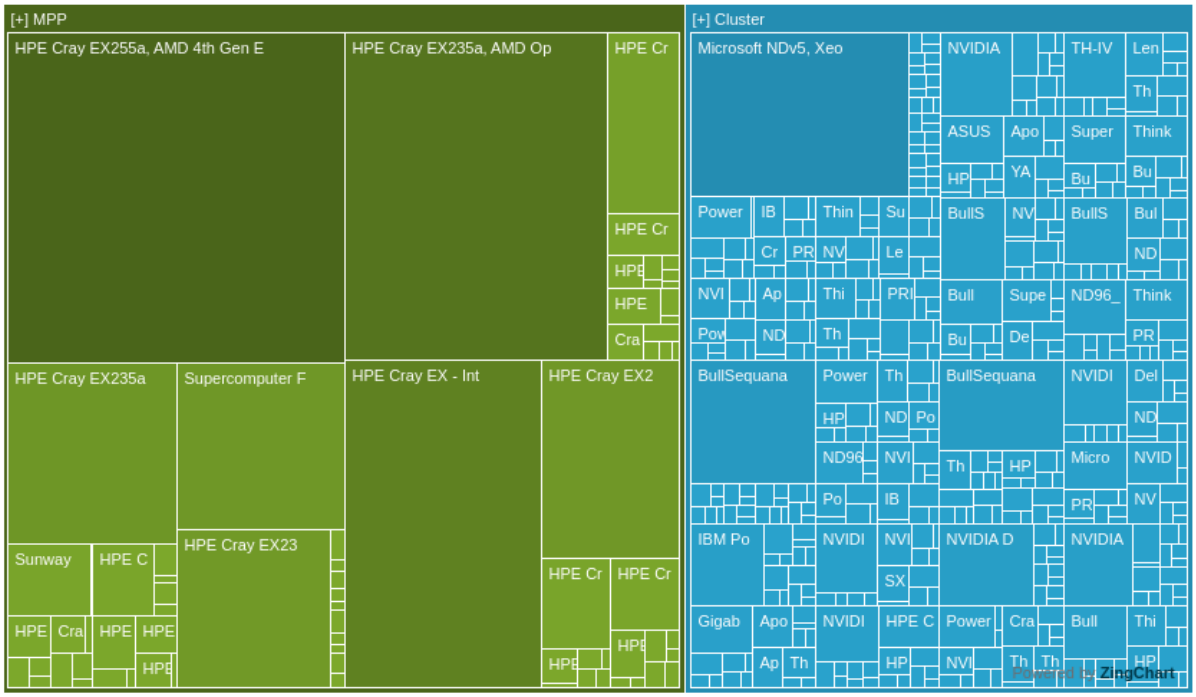


图 2: Top500 超级计算机体系结构占比情况

2.2.2 向量流水计算机的实现

以向量流水线为例，向量流水计算过程将向量的计算过程划分为多个阶段，每个阶段可以并行执行。从编程的角度，向量运算通常可以被理解为三个阶段串行执行的结果。阶段 1：加载数据，即将数据加载到向量寄存器。阶段 2：运算阶段，即执行运算，如加法、乘法等。阶段 3：存储结果，即将结果写回内存或输出寄存器。在流水线中，向量流水机可以按照上述三个阶段划分流水（详见图 3）。当有多组向量数据需要进行处理时，这种流水线技术能够确保处理器在每个时钟周期内都有数据在各级流水线中经过，从而提升计算吞吐率。

具体的流水线计算机可能会采用更多级的流水策略以进一步提高系统的并行度。Shen[15] 给出了设计向量流水线计算机的一些常见思路，其中提到了五级流水、竞争消解以及 tomasulo、记分板等流控制等相关技术。由于处理机在单周期内可以同时并行处理处于不同阶段的多条命令，多个数据在不同阶段同时执行，因而流水向量机的设计实现了事实上的指令级并行，显著提高吞吐量。通过流水线，处理器能够保持高效运作，等待时间和闲置周期减少，数据处理时延降低。

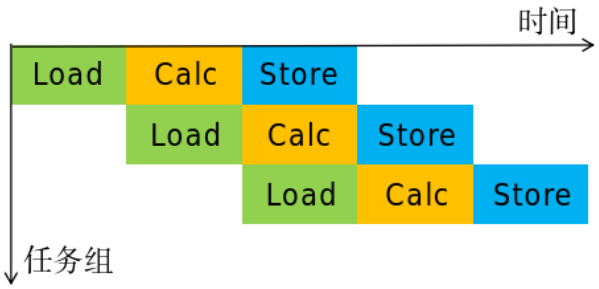


图 3: 流水线加速原理示意图

2.3 处理器阵列 SIMD

2.3.1 处理器阵列的概念

处理器阵列 SIMD 架构是另一种 SIMD 模式下的处理器加速设计 [6]。在处理器阵列 SIMD 架构中，多个处理单元（PU）以二维或一维阵列形式排列。每个 PU 都可以执行相同的操作，但每个 PU 在同一时刻处理不同的数据。与向量流水机相比，处理器阵列机中引入了更多的 PU，各个 PU 之间并行工作，从而实现吞了吐量的提升。所有参与执行同一任务的 PU 依赖一套统一的指令流的控制进行数据的批量计算。

因此，处理器阵列在执行指令时通常具有以下三个阶段：数据分发、指令同步以及结果合并。在数据分发阶段，输入数据被划分成多个块，并分别送到不同的处理单元。指令同步可以让所有处理单元并行执行相同的指令，但它们作用于不同的数据元素。在各 PU 完成相应命令后，结果合并阶段将各个处理单元的结果合并到一起，从而形成最终的计算结果。

与向量流水机相似，由于多个处理单元可以同时处理大量数据，因此处理器阵列也能够胜任很多需要大规模并行计算的任务。从设计的角度而言，相较于向量流水机基于流水阶段的并行，处理器阵列中被使用的 PU 的数量可以根据具体数据形态的需要进行调整，系统的可拓展型也更为灵活。但多 PU 并行的结构使得负载均衡问题需要额外的考虑，这种额外的考虑会增加软件与硬件的复杂性。如果 PU 调度实现不能充分利用各个 PU 的计算能力，会导致 PU 空闲，从而更容易导致计算资源的浪费。

2.3.2 处理器阵列的实现

本文主要考虑两种处理器阵列的硬件实现模式：借助 FPGA（Field-Programmable Gate Array）的 PU 实现以及 GPU 风格的 PU 实现。

GPU 的架构本质上是一个处理器阵列，其中有上千个计算单元，每个单元都执行相同的指令，处理不同的数据。这种架构适合用于处理图形渲染和科学计算等应用。GPGPU 中诸如 CUDA 以及 OpenCL 等 SIMT 编程模型为 GPU 上的并行任务执行提供了统一且便利的编程模型。与此同时，基于 warp/wavefront 的线程簇调度也能在一定程度上避免 SM（Streaming Multiprocessor）闲置导致的资源浪费。

FPGA 是一种可编程的阵列结构，它使用 LUT （Look Up Table）技术实现了对组合逻辑电路的模拟。由于其内在天然的并行性，FPGA 常被用于实现处理器阵列 SIMD（Single Instruction Multiple Threads）中的 PU 阵列。[14] 中列举了一些 FPGA 在 AI 并行加速训练中的应用实例。

Nurvitadhi[14] 认为，在很多并行计算领域中，FPGA 有着并不逊色于 GPU 的性能，但在工业实践中却更少被人提及。本文认为，GPU 的广泛普及得益于其在软件层面上的统一编程模型，这种软件层面的统一为 GPU 的商业化推广提供了便利。

2.4 SIMD 在现代处理器中的应用

Intel AVX（Advanced Vector Extensions）指令集 [1] 是 Intel 在其处理器中引入的一系列 SIMD 指令集的扩展，旨在提高处理器在处理图形、音频、视频、科学计算和金融分析等需要大量浮点运

算的应用时的性能。AVX 指令集首次出现在 Intel 的 Sandy Bridge 微架构处理器中，并于 2011 年随第二代 Core 处理器系列一同发布。

与 Intel AVX 拓展指令集类似，ARMv6 架构中引入了名为 ARM Neon 的向量指令集拓展 [2] 以实现对并行运算功能的加速。相较于 AVX 中提供的 256 位的向量寄存器，ARM Neon 提供了 128 位长度的向量寄存器。由于 ARM 指令集常用于低功耗移动设备以及嵌入式设备中，因此在环境的制约下一般无法兼顾并行需求可能带来的功耗提升。与适用于桌面端以及工作站的 x86_64 架构相比，ARM 架构有着更加明显的低功耗需求，因此高性能并不是其首要考虑。

相较于 SISD 的处理器结构，SIMD 技术通过在单一指令下并行处理多个数据元素，提高了运算效率。向量流水计算机以及处理器阵列 SIMD 是两种不同的借助 SIMD 提高计算吞吐率的思路。向量流水计算机结合了向量化处理和流水线技术，能够在多个阶段并行处理不同数据，借助阶段间的并行提高了吞吐量和效率。而处理器阵列 SIMD 通过将多个 PU 排列成阵列，借助 PU 间的并行提升 SIMD 任务的执行效率。我们可以看到两种设计思想在当今的商用 CPU 中都有着广泛体现。

3 MIMD 架构结构介绍

3.1 共享存储模型 SMP 与 CC-NUMA

在高性能计算（HPC）中，SMP（对称多处理）和 CC-NUMA（缓存一致非一致存储访问）是两种常见的内存模型。它们针对多核和多处理器的计算资源进行了优化，以应对不同的计算需求。本文中总结了两种架构的详细介绍及其各自的优缺点。

3.1.1 SMP（对称多处理器）

SMP（Symmetric Multi-Processing）是一种多处理器体系结构，其中所有处理器共享同一块物理内存。在 SMP 系统中，每个处理器在硬件上是平等的，没有主从之分，所有的 CPU 都可以访问系统中的所有内存资源。这种设计允许系统在多个处理器之间均匀分配任务，从而提高计算效率。

SMP 体系结构是一种对称的共享内存结构。对称性意味着所有处理器平等，能够平等地访问共享内存。共享内存是 SMP 体系结构的另一个关键特征，所有处理器共享同一块物理内存，内存一致性通常由硬件和软件系统共同保证。这种共享内存的特性使得进程间通信非常高效，尤其是多线程程序之间的数据交换。这使得 SMP 有利于完成需要处理器间进行频繁通信的并行任务。

由于 SMP 中各处理机在访问存储器的能力上机会均等，开发者可以利用共享内存轻松实现线程间通信与同步，编程模型比非对称的分布式内存下的编程模型简单。但也正由于各个处理器总是对所有内存地址都具有完全共享的访问能力，随着核心数目的增加，内存带宽会成为系统的主要瓶颈，这使得 SMP 主要适用于核心数较少的系统（几十个核心以内），例如中小型服务器和工作站。时常被用于多线程计算、图像处理等任务。

3.1.2 CC-NUMA（缓存一致非一致存储访问）

CC-NUMA（Cache-Coherent Non-Uniform Memory Access）是一种多处理器体系结构，其中每个处理器有自己的本地内存，且通过高速网络访问其他处理器的内存。CC-NUMA 架构的关键在于不同处理器访问内存的速度不同，访问本地内存较快，而访问远程内存则较慢。从设计哲学的角度讲，CC-NUMA 考虑到了程序与数据的局部性原理。相比于 SMP, CC-NUMA 架构中的每个处理器访问其自带的本地内存的速度非常快，但访问其他处理器相关的远程内存则有较高的延迟。不同处理器间的缓存一致性通过缓存一致性协议（如 MESI 协议）保证。

CC-NUMA 架构强调局部性，使用该架构的高性能计算机进行程序设计时，应当尽量避免频繁访问远程内存，以减访问远程存储器所带来的延迟。相比于 SMP，CC-NUMA 中的内存由于分别被不同处理器独占，随着处理器数量的增多更不容易成为整个系统的瓶颈，因此 CC-NUMA 架构相较于 SMP 而言具有更好的可拓展型。但这种非对称的设计也使得 CC-NUMA 需要采用比 SMP 更加复杂的编程模型，程序员需要额外注意内存访问局部性以及节点间的通信延迟对程序效率的影响。

CC-NUMA 架构因其优化的内存访问速度和扩展性，在需要处理大规模数据和复杂计算的场景中得到了广泛应用。这种架构常被用于大规模科学计算，例如气候模拟、粒子物理和基因组学等领域，这些领域需要对庞大的数据集进行高速处理和分析。此外，CC-NUMA 也适用于大数据分析与数据库应用，它能够有效地支持大规模数据的存储和处理，满足数据中心对高吞吐量和低延迟的需求。

3.1.3 SMP 与 CC-NUMA 的对比

表 1 从内存架构、内存一致性、可拓展性、访存延迟以及应用场景五个角度对比了 SMP 与 CC-NUMA 两种不同内存模型的异同。

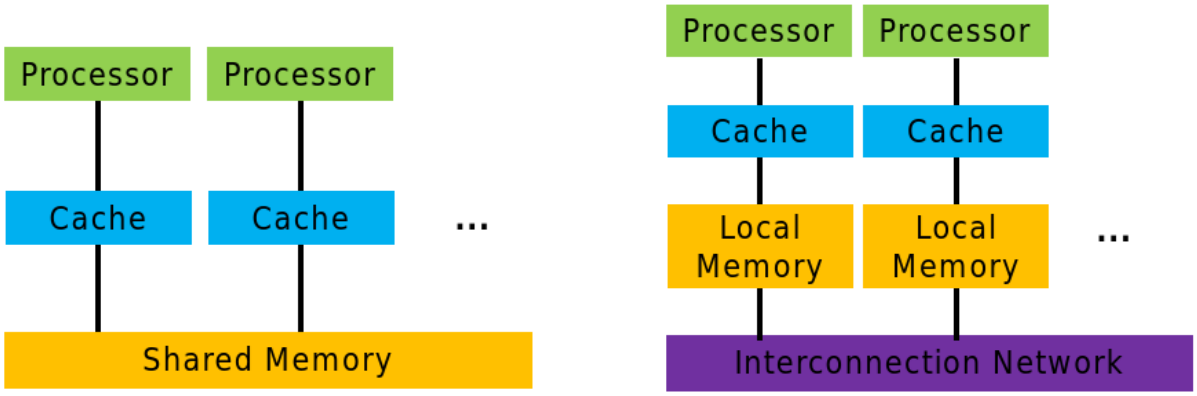
特性	SMP	CC-NUMA
内存架构	所有处理器共享同一物理内存	每个处理器拥有自己的本地内存，通过网络访问其他处理器内存
内存一致性	引入高速缓存时需要采用缓存一致性协议	一般使用诸如 MESI 的缓存一致性协议，需要考虑内存一致性
可拓展性	拓展能力有限，处理器增多时内存带宽容易成为平静	高拓展性，适合大规模计算
访存延迟	处理器间共享内存，访存通信延迟均衡	访问本地内存延迟低，访问远程内存延迟高
应用场景	中小型以及图形工作站	大规模科学计算以及数据分析

表 1: SMP 与 CC-NUMA 的对比

图 4 中给出了 SMP 以及 CC-NUMA 的典型结构对比。

3.2 PVP（并行向量处理器）

并行向量处理器的概念



(a) SMP 的典型结构　　(b) CC-NUMA 的典型结构

图 4: SMP 与 CC-NUMA 的典型结构对比

并行向量处理机（PVP）由少量的高性能专门设计定制的向量处理器 VP（Vector Processor）组成，每个至少具有 1GFLOPS 的处理能力。如图 5 所示，系统中使用了专门设计的高带宽的交叉开关网络向 VP 连向共享存储模块。并行向量处理机通过向量处理和多个向量处理器并行处理两条途径来提高处理能力。Cray X-MP、Cray T-90、NEC SX-4 和我国的银河 1 号都属于 PVP。

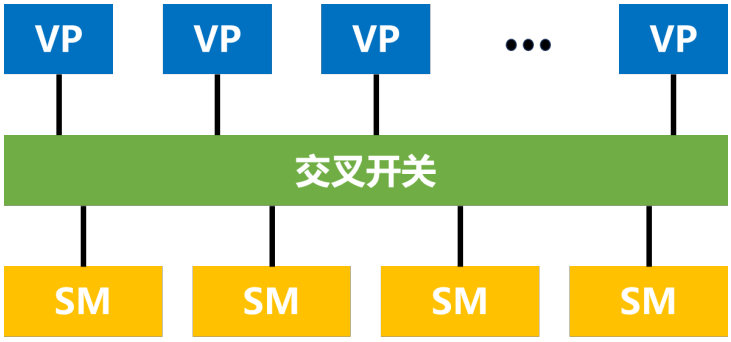


图 5: 并行向量处理器（PVP）的典型结构

PVP 通常使用定制的高带宽网络将向量处理器连向共享存储器模块。存储器可以以很高的速度向处理器提供数据。例如，在 CrayT-90 中，共享存储器能以 14GB/s 的速率将数据提供给一个处理器。这种机器通常不使用高速缓存，而是使用大量的向量寄存器和指令缓存。

CRAY X-MP

Cray X-MP 是 Cray Research 公司设计、制造和销售的超级计算机，如图 6 所示。它于 1982 年发布，是 1975 年 Cray-1 的“升级版”，并在 1983 年至 1985 年间以四处理器系统性能 800 MFLOPS 成为世界上速度最快的计算机。Cray X-MP 相对于 Cray-1 的主要改进是它是一款共享内存并行向量处理器，这是 Cray Research 推出的第一台此类计算机。它在主机中最多可容纳 4 个 CPU，主机的外观与 Cray-1 几乎完全相同。X-MP 采用新的多处理器架构，在一个机柜中容纳四个处理器。这为并行执行打开了大门，允许模型被划分为许多异步可执行的部分。这种使用模式称为多任务处理，是现代并行计算的先驱。

Cray X-MP CPU 均采用 16 门阵列集成电路。这些电路比 CRAY-1 中使用的电路更快、更密集，时钟周期时间为 9.5 ns，存储体周期时间为 38 ns。外，Cray X-MP 还采用了经过验证的冷却和封装技术，以确保系统的高可靠性。Cray X-MP 每个处理器具有四个并行内存访问端口，结合改

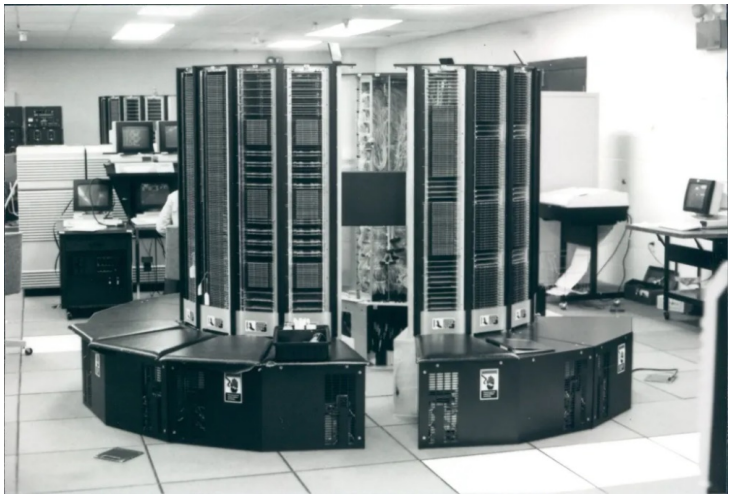


图 6: Cray X-MP

进的时钟周期，使其总可用内存带宽超过 CRAY-1 的八倍。Cray X-MP 的高性能在标量模式和向量模式下都得到了体现。标量性能通过更快的时钟、更短的内存访问时间以及更大的指令缓冲区得到提升；而向量性能则通过更快的时钟、并行内存端口以及硬件自动“链接”功能的结合得到了改进。Cray 核心结构如图 7 所示。

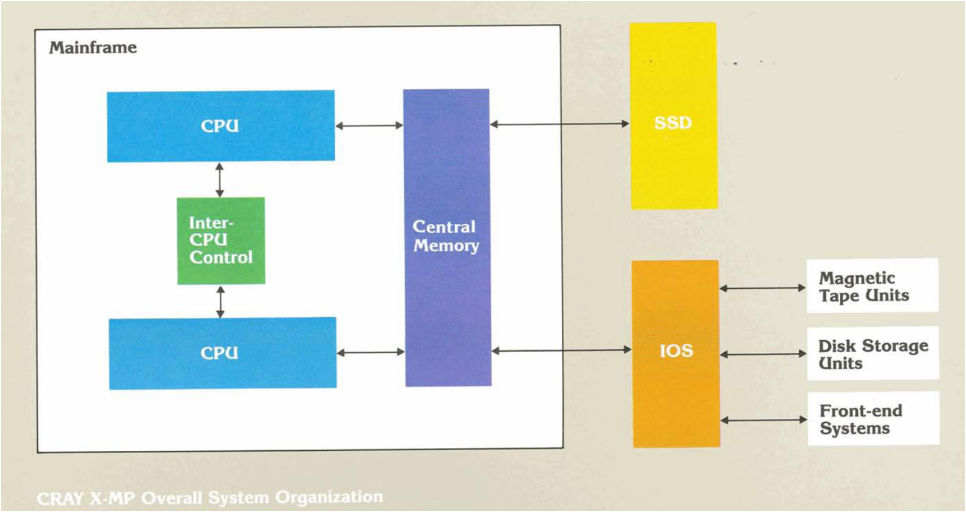


图 7: Cray X-MP

在软件层面，Cray X-MP 的创新硬件特性得到了 Cray Research 标准软件的支持。Cray 操作系统（COS）支持并发的独立单处理器任务以及单个任务的多处理。多处理可以通过 FORTRAN 程序启动和控制，同时也正在探索扩展 Cray FORTRAN 编译器（CFT）多处理能力的新技术 [10]。

在 X-MP 后，Cray Research 相继发布几款 PVP 结构的高性能处理器，其具体信息如表 2 所示 [5]。

银行 1 号

银河 1 号（银河系列）如图 8 所示高性能计算机是 1983 年中国自主研发的一款超级计算机，属于中国国家超级计算中心的高性能计算平台之一。银河 1 号在设计和性能上具有一定的创新性和领先性，主要用于科学计算、气象预测、工程模拟、数据处理等领域。

发布时间	型号	最大 CPU 数	每个 CPU 的 CPU 速度大约峰值
1976	C1	1	0.160 GFLOPS
1982	XMP	4	0.235 GFLOPS
1988	YMP	8	0.333 GFLOPS
1991	C90	16	1 GFLOPS
1995	T90	32	2 GFLOPS

表 2: Cray 系列并行处理器性能表

图 8: 银河一号

NEC SX-4

1994 年 11 月，NEC 在全球推出 SX-4 系列超级计算机如图 9 所示，其最大向量性能可达 1 teraFLOPS（每秒进行一万亿次浮点运算），采用 CMOS 和并行处理技术，性能卓越。在最大配置下，SX-4 配备了 512 个并行 CPU，其向量性能可达到惊人的 1 teraFLOPS。这一成果得益于高密度超快 CMOS LSI（几何尺寸为 0.35 微米，集成约 400 万个晶体管）和高速 4 M bit 同步 SRAM 的应用。

并行向量处理机未来发展趋势

并行向量处理机在高性能计算（HPC）领域具有显著优势。其强大的向量处理能力使得它能够高效处理大规模数据集，特别适用于科学计算、气候模拟、物理仿真和基因组学等计算密集型任务。通过对向量化操作的优化，PVP 能够实现极高的计算吞吐量，极大缩短计算时间。此外，PVP 还具备优异的并行性，能够通过多个处理单元同时执行复杂计算，充分发挥现代多核处理器的优势。

尽管并行向量处理机在许多领域表现出色，但其发展也面临一些不足和挑战。首先，PVP 系统通常对程序的向量化优化要求较高，需要专门的编程技术和工具支持，这可能增加了开发和维护的复杂性。此外，PVP 的硬件成本较高，尤其是在大规模部署时，能耗和散热问题也成为制约其发展的瓶颈。

从应用的角度看，PVP 通常需要与其他计算平台（如 GPU、FPGA 等）协同工作，构建异构计算系统，这要求开发人员具备更多的硬件与软件整合能力。

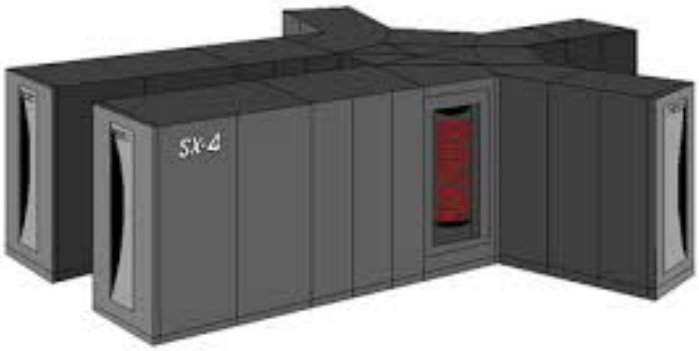


图 9: NEC SX-4

3.3 MPP（大规模并行处理器）

MPP（Massively Parallel Processing）是一种基于分布式计算的体系结构，旨在通过大量独立的计算节点并行执行计算任务。其核心思想是将一个大的数据处理任务分解为多个小的子任务，并在多个处理器上同时执行这些子任务，以实现高效的并行处理。每个节点都有独立的磁盘存储系统和内存系统，业务数据根据数据库模型和应用特点划分到各个节点上，每台数据节点通过专用网络或者商业通用网络互相连接，彼此协同计算。MPP 具有完全的可伸缩性、高可用、高性能、优秀的性价比、资源共享等优势。

MPP 的节点互联机制与 NUMA 不同。在单台物理服务器内部，MPP 通过高速互联实现处理器与本地内存的非一致性访问，并通过网络互联实现多个节点的分布式计算。其中，每个节点通常是一个独立的 SMP 系统。节点互联通常由专用硬和高速通信网络组成。这些计算节点通常拥有独立的内存，但节点间的通信延迟较低，通常采用专用的高速网络（如 InfiniBand）来减少通信开销。

MPP 将大型数据处理任务分解为多个小的子任务，并在多个节点上并行执行，从而显著提升数据处理速度。同时，数据在各个节点上分布式存储，使得每个节点能够独立处理其数据，并与其他节点协同工作。此外，MPP 架构采用分布式计算方式，每个节点独立计算一部分数据，最终汇总结果，这不仅提高了计算效率，还减少了单点故障的风险。其高并发性使得单个节点可以支持超过 300 个用户的并发访问，能够高效处理大规模数据集。随着数据规模的增长，MPP 架构支持通过增加节点数量进行横向扩展，展现出良好的可扩展性。最后，MPP 采用 Shared Nothing（完全无共享）架构，每个节点拥有独立的 CPU、内存和其他硬件资源，确保了系统的稳定性和可靠性。

3.4 Cluster（集群）

计算集群通常指的是由多台独立的计算机（节点）组成的一个集合。这些计算机通过网络互联，共享任务来完成并行计算。计算集群的计算资源可以是同质的（所有节点具有相同硬件）或异质的（不同节点可能具有不同硬件配置）。集群系统通常使用分布式内存或共享内存（取决于集群的实现）。集群中的计算节点可以是低成本的标准服务器，通常配置为运行 Linux、Windows 或其他操作系统。

节点互联集群系统通常依赖于标准硬件，而不是专用硬件。资源可能分散在多个物理机上，这些物理机通过标准网络连接（如千兆以太网、万兆以太网或 InfiniBand）进行通信。集群的资源管理通常依赖于操作系统和调度软件（如 Slurm、PBS、Torque、Hadoop 等）。计算集群的通信可能不像 MPP 那样具备专用的硬件支持，从而导致通信延迟相对较高，特别是在大规模集群中。

Cluster 的主要优点在于成本低，它基于普通的商用硬件，性价比高。同时，集群具有较高的灵活性，能够轻松扩展或替换节点，适应不断变化的需求。此外，集群的容错性也很强，单个节点的失效不会导致整个系统的瘫痪。然而，集群的管理复杂性较高，尤其是在大规模集群中，管理和维护任务包括处理节点故障、负载均衡和资源调度等，增加了运维难度。此外，集群中节点之间的通信依赖于网络带宽和延迟，在大规模集群中，通信开销可能成为瓶颈，影响整体性能。同时，在分布式存储系统中，如何确保数据的一致性和完整性也是一个重要问题，特别是在节点之间频繁进行数据交换的情况下，这为系统的稳定性带来了挑战。

3.5 MPP 与 Cluster 的对比

两者的区别在于，MPP 系统往往设计上更加集成、专用，追求极致的并行计算性能，而集群系统则更多依赖通用计算节点，通过灵活的资源管理实现分布式计算，适用范围更广。表 3 给出了一个更具体的 MPP 与 Cluster 的特性对比。

特性	MPP	Cluster
架构	大规模并行计算专用硬件，紧密的集成结点，优化通信	由多台独立的计算机组成，资源可以共享或者独立
资源管理	专用硬件，优化的资源调度和任务调度	标准硬件，有很多灵活的资源调度工具（Slurm、PBS 等）
通信方式	高效的告诉网络（如 InfiniBand）连接，低延迟通信	通过标准网络连接，通信开销更大，依赖于调度软件
扩展性	有限，增加节点需要专门的硬件支持	良好，扩展非常灵活，增加节点较为容易
应用场景	高性能计算、科学计算、工程计算等	大数据处理、Web 服务、虚拟化、大规模并行任务等

表 3: SMP 与 CC-NUMA 的对比

4 总结

表 4 汇总了本文调研各类体系结构时所收集到的具体的超算实例信息以及相关参考文献，图 10 给出了这些信息的一个相对直观的展示。分析超级计算机的历史发展脉络，我们不难发现，在超级计算机出现早期，研究工作核心关注于 PVP 系统的设计。PVP 系统致力于构建高性能的单机向量处理系统。而随着单芯片集成度逐渐达到量子极限，研究工作的重心逐渐从高效的单机系统向多机协同系统转移。

从客观上来讲，多种力量正促进着这种转移的出现。一方面，商业化芯片的日臻成熟促成了一大批 Cluster 集群步入人们的视野，构建多机系统的成本逐渐降低，这使得设计专用的硬件产

品显得过于费时费力。另一方面，计算机网络的发展也进一步促进了分布式系统的发展与完善，使得原先难以被分布式完成的任务，可以在相对高效的光纤以太网中合理且高效地完成数据分发工作。

但调度、通信以及多机管理对于 Cluster 在性能领域的长足进步仍然颇有限制。出于这种限制，我们可以推测，MPP 与 Cluster 在超算领域“平分天下”的格局在短时间内很难有显著的变化。MPP 因其极高的性能在科研领域发挥重要作用，为理论与工程研究提供重要支撑，但由于其硬件的专用性以及资源的集中性，因而常常不利于市场化与商业化的普及。而 Cluster 系统依托各大软件厂商的云服务平台，为其他企业或个人提供易于灵活拓展的计算服务。

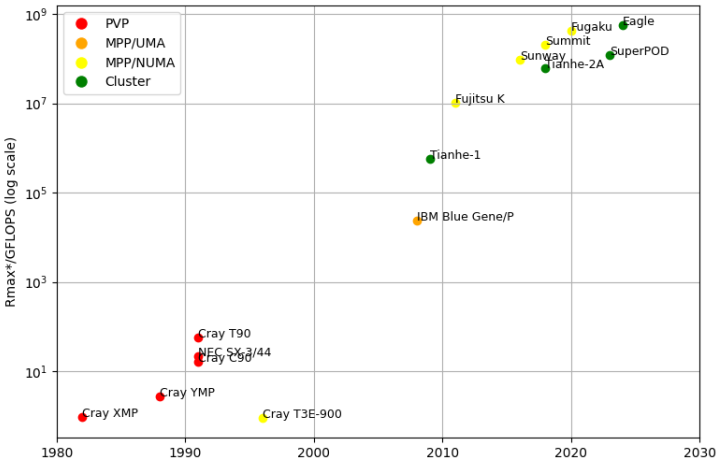


图 10: 文中提及的所有高性能计算机的性能图表

名称	架构分类	建成年份	等效 Rmax	数据来源
Cray XMP	PVP	1982	0.235 GFLOP × 4^-	[5]
Cray YMP	PVP	1988	0.333 GFLOP × 8^-	[5]
Cray C90	PVP	1991	1 GFLOP × 16^-	[5]
Cray T90	PVP	1991	1.8 GFLOP × 32^-	[5]
NEC SX-3/44	PVP	1991	22 GFLOPS	[23][13]
Tianhe-1	Cluster	2009	563.10 TFLOPS/s	[24]
Tianhe-2A	Cluster	2018	61.44 PFLOPS	[25]
Eos NVIDIA DGX SuperPOD	Cluster	2023	121.40 PFLOPS	[18]
Eagle	Cluster	2023	561.2 PFLOPS	[17]
Cray T3E-900	MPP/NCC-NUMA	1996	0.9 GFLOPS	[3]
IBM Blue Gene/P	MPP/UMA	2008	23.86 TFLOPS	[16]
Fujitsu K	MPP/Tofu(NUMA)	2011	10.5 PFLOPS	[19]
Fugaku	MPP/CC-NUMA	2020	442.01 PFLOPS	[22]
Summit	MPP/Hybrid(NUMA)	2018	200 PFLOPS	[20]
Sunway TaihuLight	MPP/CC-NUMA	2016	93.01 PFLOPS	[21]

表 4: 超算信息及其参考资料

5 分工情况以及参考文献

5.1 小组分工情况

小组成员	学号	负责内容
宋震	ZB2406316	SIMD：向量流水机、阵列处理器
苏晓	BY2406130	SMP、CC-NUMA
张泰然	ZY240633	MIMD：PVP
刘旭昭	SY2406327	MIMD：MPP、Cluster
郭冠男	SY2406410	编写概述，汇总，制作 PPT 及文档

表 5: 分工表

参考文献

[1] Wikipedia contributors. *Advanced Vector Extensions*. Accessed: 2024-11-30. 2024. URL: https://en.wikipedia.org/wiki/Advanced_Vector_Extensions.

[2] Wikipedia contributors. *ARM architecture family*. Accessed: 2024-11-30. 2024. URL: https://en.wikipedia.org/wiki/ARM_architecture_family.

[3] Wikipedia contributors. *Cray T3E*. Accessed: 2024-11-30. 2024. URL: https://en.wikipedia.org/wiki/Cray_T3E.

[4] Wikipedia contributors. *Seymour Cray*. Accessed: 2024-11-30. 2024. URL: https://en.wikipedia.org/wiki/Seymour_Cray.

[5] Cray-History.net. *PVP Generations, XMP, YMP, C90, T90*. Accessed: 2024-11-30. 2024. URL: https://cray-history.net/faq-1-cray-supercomputer%20families/#TOC3.

[6] David Culler, Jaswinder Pal Singh, and Anoop Gupta. *Parallel Computer Architecture: A Hardware/Software Approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. ISBN: 9780080573076.

[7] M.J. Flynn. "Very high-speed computing systems". In: *Proceedings of the IEEE* 54.12 (1966), pp. 1901–1909. DOI: 10.1109/PROC.1966.5273.

[8] Yong Fu. *Performance Optimization on Modern Processor Architecture through Vectorization*. Accessed: 2024-11-30. 2016. URL: https://objectcomputing.com/resources/publications/sett/december-2016-performance-optimization-on-modern-processor-architecture-through-vectorization.

[9] Daniel Gerzhoy et al. "Nested MIMD-SIMD Parallelization for Heterogeneous Microprocessors". In: 16.4 (Dec. 2019). ISSN: 1544-3566. DOI: 10.1145/3368304. URL: https://doi.org/10.1145/3368304.

[10] cray-rearch inc. *The CRAY X-MP Series of Computers*. Accessed: 2024-11-30. 1983. URL: http://s3data.computerhistory.org/brochures/cray.x-mp.1983.102646267.pdf.

[11] Norman P Jouppi et al. “In-datacenter performance analysis of a tensor processing unit”. In: *Proceedings of the 44th annual international symposium on computer architecture*. 2017, pp. 1–12.

[12] Norman P Jouppi et al. “Ten lessons from three generations shaped google’ s tpuv4i: Industrial product”. In: *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE. 2021, pp. 1–14.

[13] museum.ipsj.or.jp. 【*NEC*】*SX-4*. Accessed: 2024-11-30. 2024. URL: https://museum.ipsj.or.jp/en/computer/super/0018.html.

[14] Eriko Nurvitadhi. *Real Performance of FPGAs Tops GPUs in the Race to Accelerate AI*. Accessed: 2024-11-30. 2020. URL: https://www.intel.com/content/dam/www/central-libraries/us/en/documents/real-performance-of-fpgas-tops-gpus-in-the-race-to-accelerate-ai-whitepaper.pdf.

[15] John Shen and Mikko Lipasti. *Modern Processor Design: Fundamentals of Superscalar Processors*. Morgan Kaufmann, 2005.

[16] top500.org. *Blue Gene/P Solution*. Accessed: 2024-11-30. 2008. URL: https://www.top500.org/system/176518/.

[17] top500.org. *Eagle - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR*. Accessed: 2024-11-30. 2023. URL: https://top500.org/system/180236/.

[18] top500.org. *Eos NVIDIA DGX SuperPOD - NVIDIA DGX H100, Xeon Platinum 8480C 56C 3.8GHz, NVIDIA H100, Infiniband NDR400*. Accessed: 2024-11-30. 2023. URL: https://www.top500.org/system/180239/.

[19] top500.org. *K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect*. Accessed: 2024-11-30. 2011. URL: https://top500.org/system/177232/.

[20] top500.org. *Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband*. Accessed: 2024-11-30. 2018. URL: https://top500.org/system/179397/.

[21] top500.org. *Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway*. Accessed: 2024-11-30. 2016. URL: https://www.top500.org/system/178764/.

[22] top500.org. *Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D*. Accessed: 2024-11-30. 2020. URL: https://www.top500.org/system/179807/.

[23] top500.org. *SX-3/44*. Accessed: 2024-11-30. 1997. URL: https://www.top500.org/system/170812/.

[24] top500.org. *Tianhe-1 - NUDT TH-1 Cluster, Xeon E5540/E5450, ATI Radeon HD 4870 2, Infiniband*. Accessed: 2024-11-30. 2008. URL: https://top500.org/system/176546/.

[25] top500.org. *Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000*. Accessed: 2024-11-30. 2018. URL: https://top500.org/system/177999/.

[26] 中存储. 从超算榜 *TOP500* 看 *Cluster* 和 *MPP* 架构在超算系统中的实际应用. Accessed: 2024-11-30. 2018. URL: https://wap.chinastor.com/hpc-top500/06303SD2018.html.