## High Performance Computing- History of the Supercomputer

早期

1943 Colossus 是第一台可编程的数字电子计算机，二战时期为破译密码而秘密设计，对后世计算机影响不大

1945 Manchester Mark I 意义重大，它是第一台使用索引寄存器修改基址的机器

1950 MIT Whirlwind是第一台实时运行的计算机，使用视频显示进行输出(之前的都是批处理计算机)，为了应对这种需求，创建了核心存储器（在磁场极性中存储数据的铁氧体环）

1960 LARC 是第一台真正的超级计算机，它是为多处理而设计的

1961 7030 IBM为了击败LARC设计的计算机，有很多创造性举措得以沿用，例如多重编程、内存保护、广义中断、8 位字节、指令流水线、预取和解码以及内存交错

1965 CDC 6600 比当时的最快计算机快10倍，第一个RISC系统，具备10个外围处理器来处理IO和运行操作系统

向量时代

1974 CDC Star-100 是最早使用矢量处理器的机器之一，为了提高矢量性能，牺牲了基本的标量性能，机器通常被认为是失败的。

1975 Cray-1 使用矢量寄存器而不是流水线内存操作而不影响标量性能的矢量处理器。附带Cray操作系统、Cray 汇编语言 和 Cray FORTRAN，第一个自动矢量化的 FORTRAN 编译器

1981 CDC Cyber-205 纠正了 Star-100 的错误，使用半导体内存和虚拟内存概念

1983 Cray X-MP 并行矢量处理器机器，更好的链接支持、并行算术流水线和共享内存访问，每个处理器具有多个流水线。

传统时代 1991-2010

1995 Intel ASCI Red 2.15 TFLOPS

2000 IBM ASCI White 7.226 TFLOPS

2002 Earth Simulator 35.86 TFLOPS

2005 IBM ASCI Blue Gene 70 -478 TFLOPS

2008 IBM Roadrunner 1.105 PFLOPS

2009 Cray Jaguar 1.75 PFLOPS

GPU时代 2011-至今

2010 Tianhe-1A 2.57 PFLOPS

2011 Fujitsu K computer 8.2 –10.5 PFLOPS

2012 IBM Sequoia 20.1 PFLOPS

2013 Tianhe-2 54.9 PFLOPS

2015 Sunway TaihuLight 125.4 PFLOPS

2018 Summit 187.6 PFLOPS

2020 Fugaku 415.5 PFLOPS

2022 Frontier 1102 PFLOPS

## The CRAY-1 Computer System

1976

历史上最成功的超级计算机，能够在持续周期内支持每秒1.38亿次浮点运算(MFLOPS)的计算速率。

通过一种叫做链接的技术，CRAY-1矢量功能单元与标量和矢量寄存器相结合，产生临时结果并立即再次使用它们，而不需要额外的内存引用，这在其他当时的计算机系统中减慢了计算过程。

使用链接技术，当从一个功能单元发出的结果(以一个/时钟周期的速率)立即馈送到另一个功能单元，以此类推。换句话说，中间结果不必存储到内存中，甚至可以在创建它们的vector操作运行完成之前使用。

链接类似于IBM计算机中使用的“数据转发”技术。与数据转发一样，链接也是自动进行的。

增强CRAY-1的计算能力的特点是:它的体积小（基座约0.5米高直径约2.5米，主体不到2米直径约1.5米），这减少了电信号必须在计算机框架内传播的距离，并允许12.5纳秒的时钟周期，拥有当时世界上最快的标量处理器。具有错误检测和纠错逻辑的100万字半导体存储器(SECD~D)，具有64位字长，及其优化的Fortran编译器。

## Reevaluating Amdahl's Law

1988

阿姆达尔定律：

优化前用时T1，优化后用时Tn，记F为程序中的串行比例，则(1-F)是并行比例，由此可得加速比为T1/Tn，将Tn=T1(F+(1-F)/n)代入化简得加速比s=1/(F+(1-F)/n)

可以从“加速比”的公式中看出，单纯地增加处理器的数量并不一定可以有效地提高系统的性能，只有在提高系统内并行化模块比重的前提下，同时合理增加处理器的数量，才能以最小的投入得到最大的加速比

作者基于此提出了古斯塔夫森定律：

记a为串行时间，b为并行时间，n为处理器个数，则加速比s=(a+nb)/(a+b)，由于串行比例F=a/(a+b)，变换后得到s=n-F(n-1)

可以看出，F（串行化程度）足够小，也即并行化足够高，那么加速比和cpu个数成正比。也是说明处理器个数、串行比例和加速比之前的关系，只不过它的侧重角度有所不同。

这两个定律是高并发程序中的重要定律，对并行计算的理论基础有重要贡献。

## What’s Next in High-Performance Computing

2002

超级计算机存在两种主要架构:克雷式向量超级计算机集群;以及标量单和多处理器集群。集群正在从运行专有软件的大规模并行计算机和集群过渡到运行标准软件的专有集群，以及从商用硬件和软件构建的自己动手的Beowulf集群。

它建立在几十年的并行处理研究和将松耦合计算机应用于各种应用的许多尝试之上。部分组件包括:消息传递接口(MPI)编程模型;并行虚拟机(PVM)编程，执行和调试模型;并行文件系统;配置、调度、管理和调整并行应用程序的工具;高级库，例如Linpack, BLAS。

Beowulf商品集群的缺点是，在需要大量共享内存的应用程序上表现不佳。

## High-Performance Computing: Clusters, Constellations, MPPs, and Future Directions

2005

基于上一篇文章给出的集群概念进行完善和修正，集群的定义范围限制为由独立节点的集成集合组成的并行计算机系统，每个节点都是一个独立的系统，能够独立运行，并且派生于为其他独立目的开发和销售的产品

例如，集群- now系统几乎完全使用消息传递接口(MPI)编程，而星座系统可能至少部分使用OpenMP编程，使用线程模型。通常，星座是空间共享的，而不是时间共享的，每个用户都有自己的节点;空间共享clusterNOW系统意味着为特定用户分配一定数量的节点。

强调描述并行计算系统的四个主要维度:集群、命名空间、并行性以及延迟和局部性管理

## Distributed Shared Memory: Concepts and Systems

1996

多处理器系统的内存系统组织，通常分为两大类:共享内存系统和分布式内存系统。

共享内存系统(通常称为紧耦合多处理器)使所有处理器都可以平等地访问全局物理内存。

具有简易性和可移植性。但是，共享内存多处理器在访问共享内存时通常会遇到争用增加和延迟延长的问题。分布式存储系统(通常称为多计算机)由多个独立的处理节点和本地存储模块组成，通过通用互连网络连接。由于必须的处理跨系统的数据分布和管理通信，以及进程迁移也会带来问题。因此，与共享内存系统相比，分布式内存系统中的硬件问题更容易，软件问题更复杂。

分布式共享内存结合了这两种方法的优点。

DSM系统在物理分布式内存系统上逻辑地实现共享内存模型。系统设计人员可以通过各种方式在硬件或软件中实现实现共享内存抽象的特定机制。

DSM系统一般结构DSM系统一般由一组节点或集群组成，通过互联网络连接。集群本身可以是单处理器或多处理器系统，通常围绕共享总线组织。

用于分发共享数据的两种常用策略是复制和迁移。复制允许同一数据项的多个副本驻留在不同的本地内存(或缓存)中。它主要用于使不同站点能够同时访问相同的数据，主要是在读取共享盛行的情况下。迁移意味着在任何时候只存在数据项的一个副本，因此必须将数据项移动到请求站点以独占使用。为了减少一致性管理开销，在普遍采用顺序写共享模式时，用户更喜欢这种策略。系统设计人员必须选择一种DSM算法，这种算法必须很好地适应典型应用程序中内存引用的系统配置和特征