并行与分布式计算基础：第一讲

杨超
chao_yang@pku.edu.cn

2019 秋

内容提纲

课程介绍

课程基本信息

- 课程名称：并行与分布式计算基础
- 英文名称：Foundations of Parallel and Distributed Computing
- 授课教师：杨超（chao_yang@pku.edu.cn，理科 1 号楼 1520）
- 课程助教：尹鹏飞（pengfeiyin@pku.edu.cn）
- 学分：3，周学时：3，总学时：51
- 授课对象：数据科学专业、应用统计专业、其他相关专业
- 先修课程：计算概论、数据结构、机器学习等
- 考核方式：平时作业 (50%) + 期末考试 (50%)
- 教材：无

课程定位

基本定位

- 在过去的十年里，并行与分布式计算的需求正经历爆炸式增长，已经从一门选修课程变成了计算科学和数据科学课程体系的核心组成部分；
- 本课程包含关于并行与分布式计算的计算模型理论的基本概念和分布式内存体系架构上的 MPI 编程技术、共享内存体系架构上的 OpenMP 编程技术以及在 GPU 众核体系架构上的 CUDA 编程技术等；
- 通过本课程的学习，将对并行与分布式计算的基础理论、编程方法及其在计算科学和数据科学中的应用有较为系统性的了解，从而提高算法设计、编程与应用等相关能力。

授课内容

主要内容（暂定）

- 引言
- 硬件体系架构
- 并行计算模型
- 编程与开发环境
- MPI 编程与实践
- OpenMP 编程与实践
- GPU 编程与实践
- 前沿问题选讲

上课时间（地点：二教 211）

上课时间	星期一	星期二	星期三	星期四	星期五
第 1 节 (8:00-8:50)					
第 2 节 (9:00-9:50)					
第 3 节 (10:10-11:00)				单周	
第 4 节 (11:10-12:00)				单周	
第 5 节 (13:00-13:50)		每周			
第 6 节 (14:00-14:50)		每周			
第 7 节 (15:10-16:00)					
第 8 节 (16:10-17:00)					
第 9 节 (17:10-18:00)					
第 10 节 (18:40-19:30)					
第 11 节 (19:40-20:30)					
第 12 节 (20:40-21:30)					

联系方式

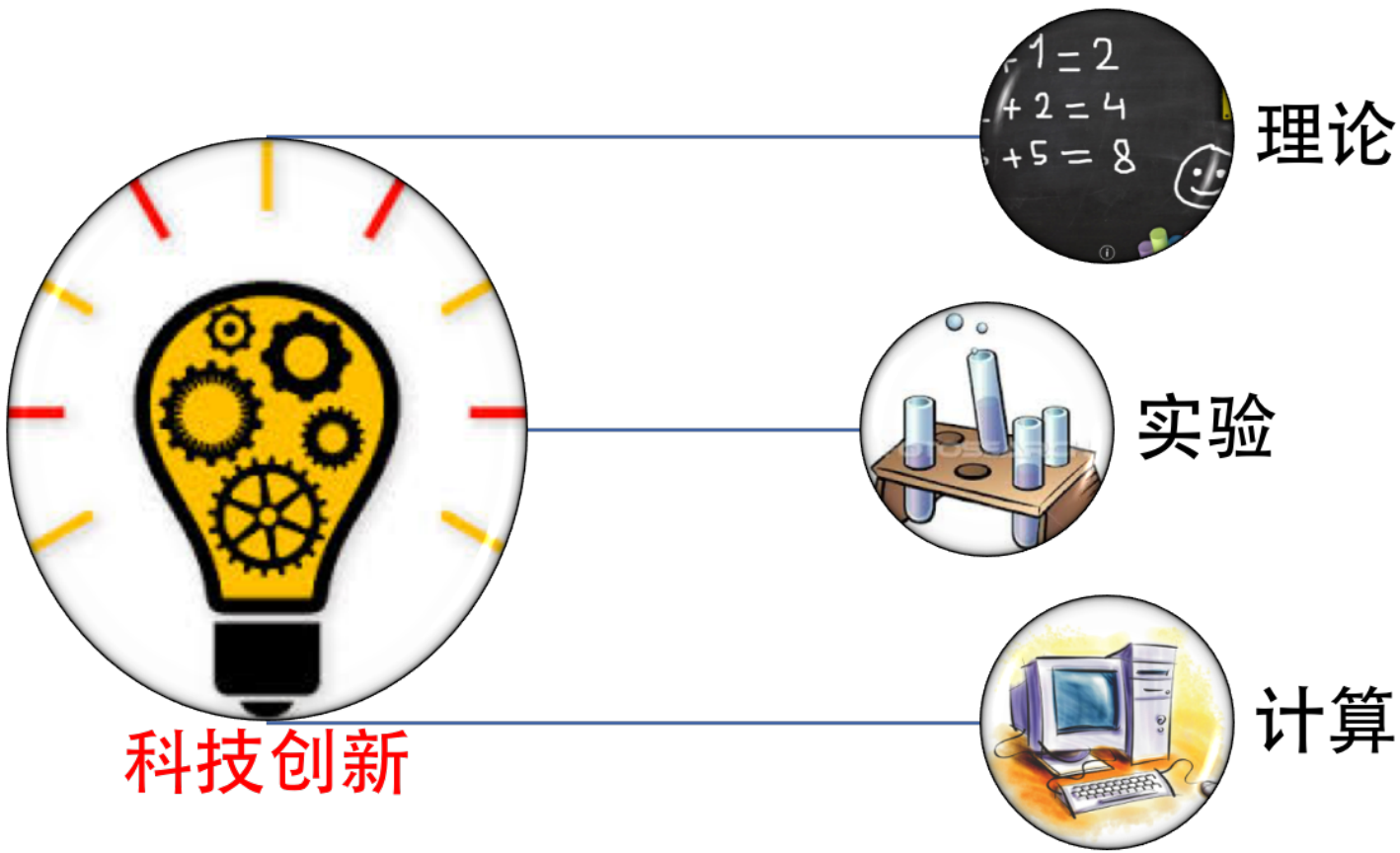
并行与分布式计算基础2019

该二维码7天内(9月16日前)有效，重新进入将更新

引言

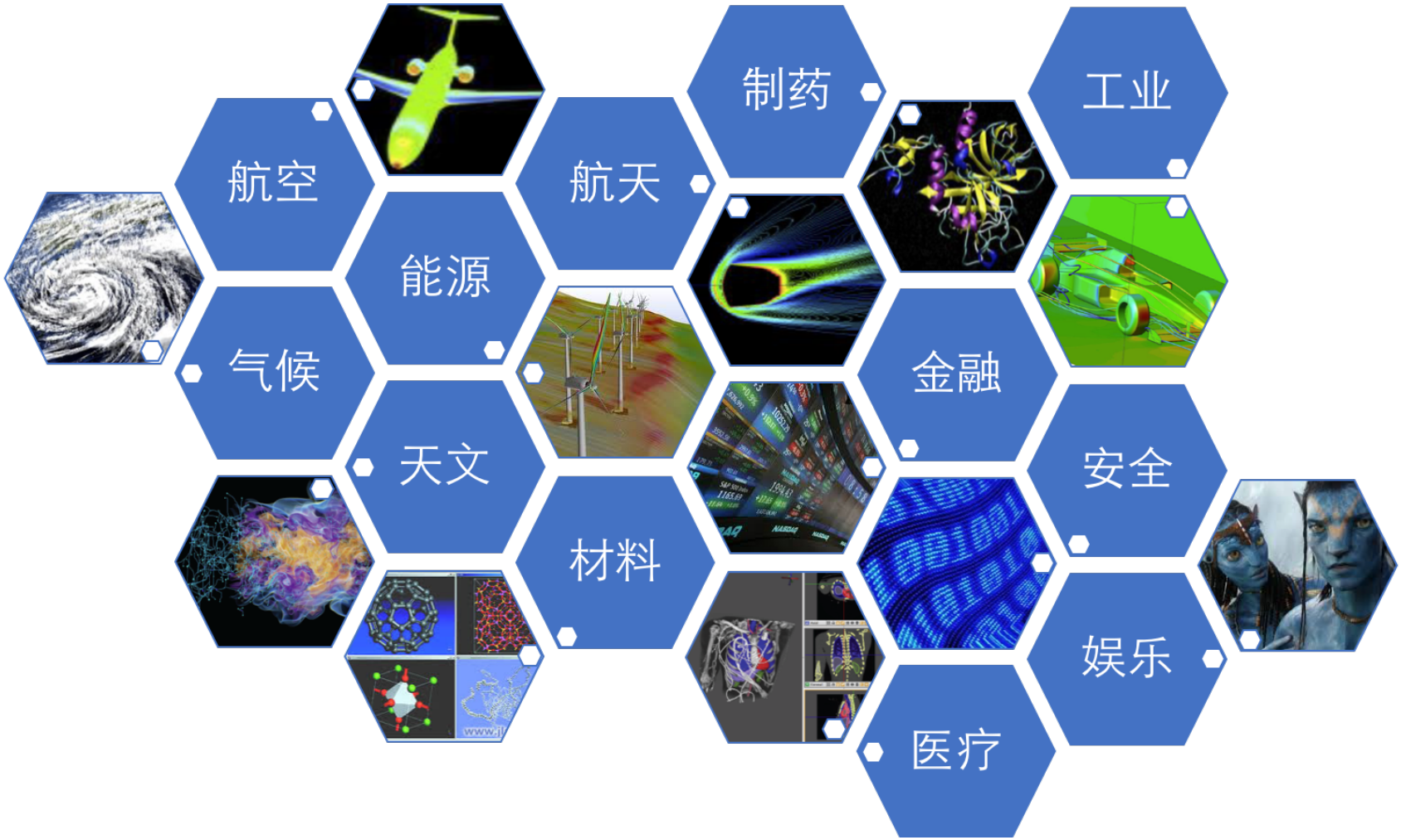
计算的重要性

- 计算已经成为科技创新的第三大手段



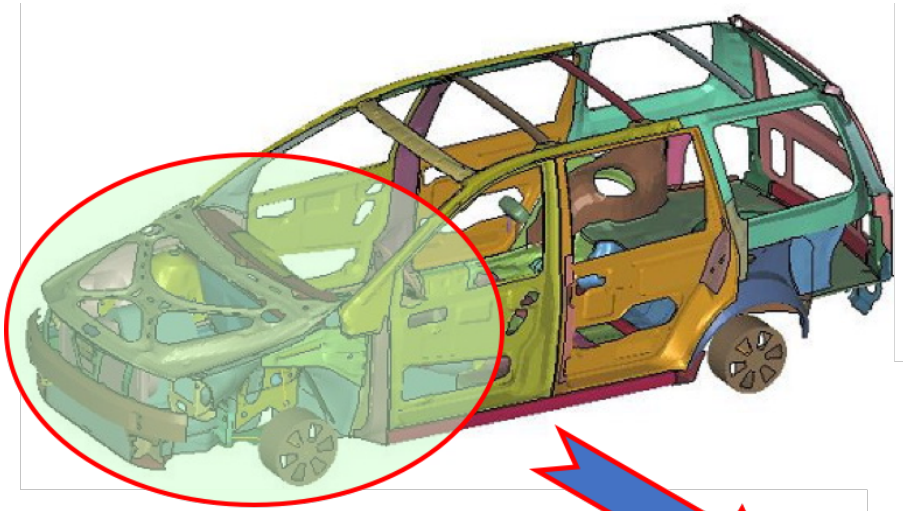
计算的普遍性

- 计算已经在各行各业中大显身手

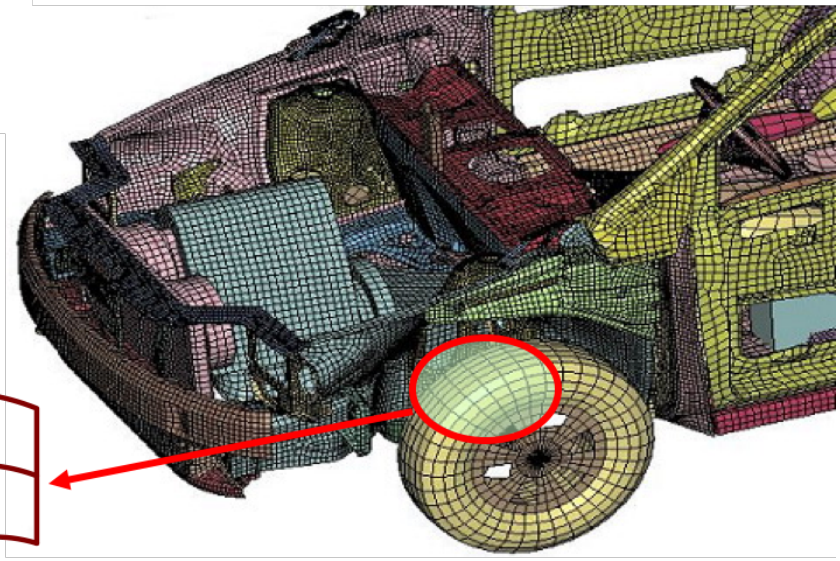


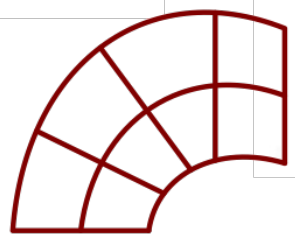
举例：汽车碰撞仿真

- 每一种车型的汽车出厂前都需要进行大量安全性测试，其中，通过计算机进行汽车碰撞的数值仿真实验，可以相当程度地代替真车实验，大幅度节省开支。

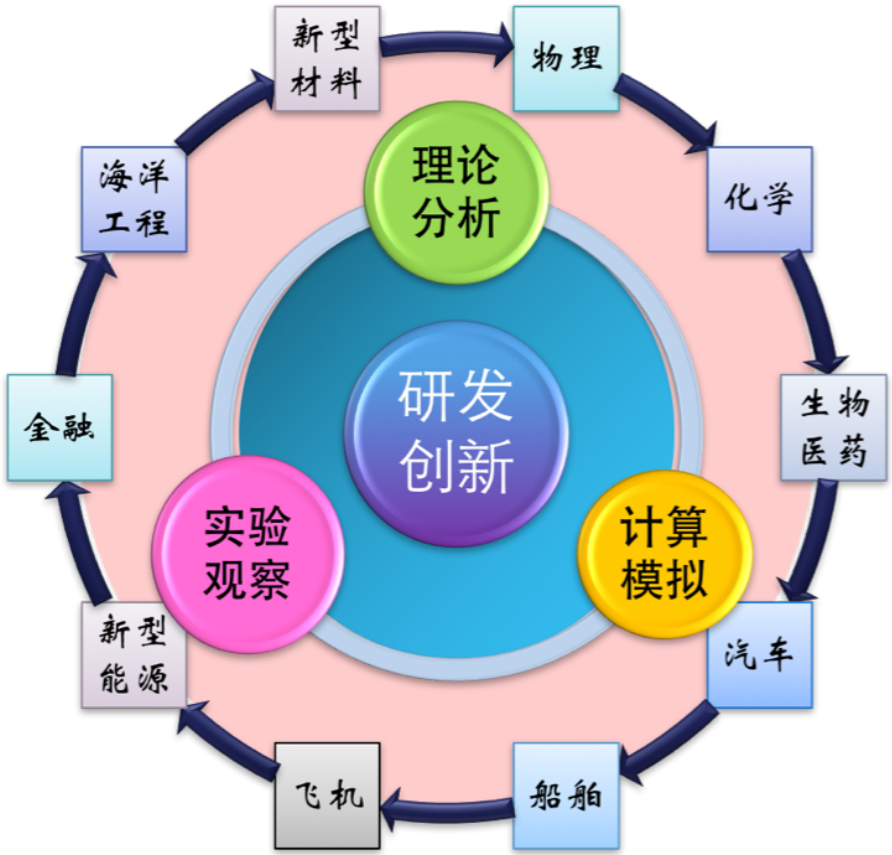


$$\frac{\mathrm{d}}{\mathrm{d}t}\left\{\frac{\partial E_{\mathrm{K}}}{\partial \dot{z}_i}\right\}-\frac{\partial E_{\mathrm{K}}}{\partial z_i}+\frac{\partial E_{\mathrm{P}}}{\partial z_i}+\frac{\partial E_{\mathrm{D}}}{\partial \dot{z}_i}=F_{z_i},\quad i=1,2,\cdots,8,$$





计算的复杂性



应用领域的强烈需求

- 人口愈来愈多
- 数据愈来愈大
- 节奏愈来愈快
- 灾害愈来愈频
- 雾霾愈来愈重
- ...

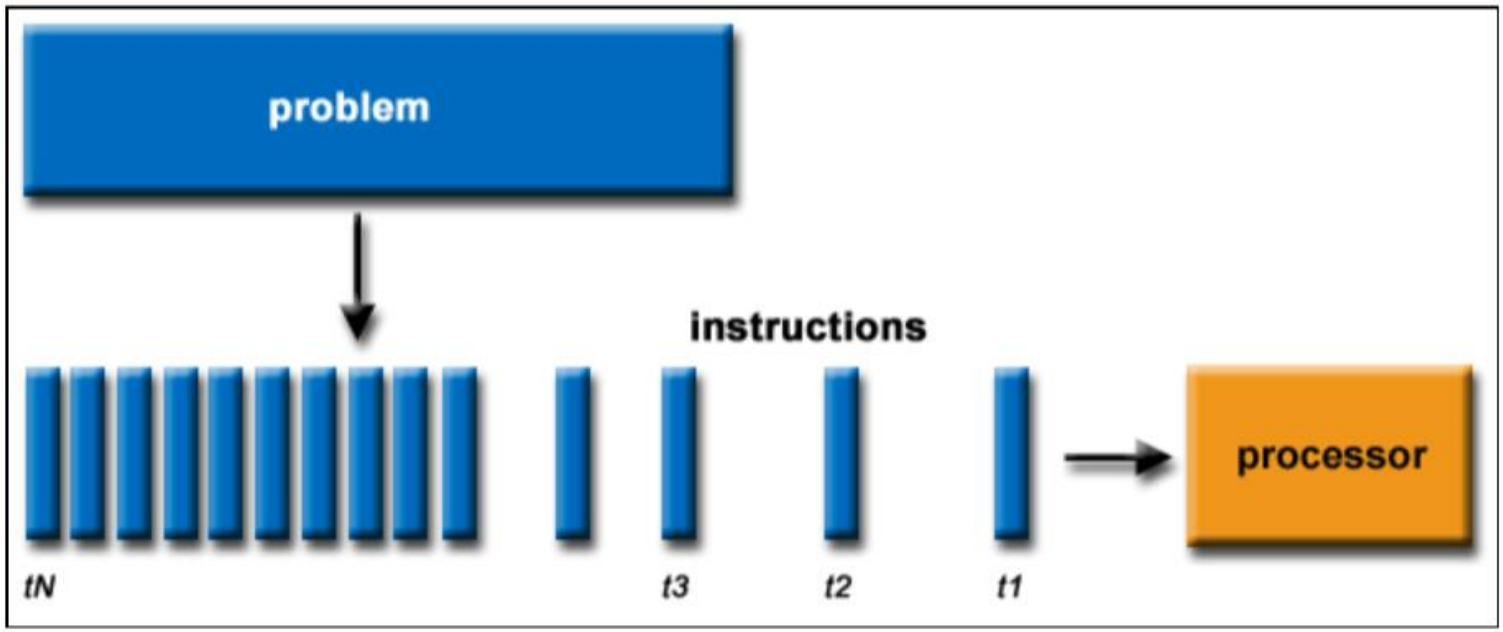
串行计算 vs 并行计算

- 串行计算将问题被分为一系列独立指令，按照先后顺序逐一执行；
- 并行计算则将问题分为能并发执行的若干部分，分别串行执行。

什么是并行计算?

串行计算

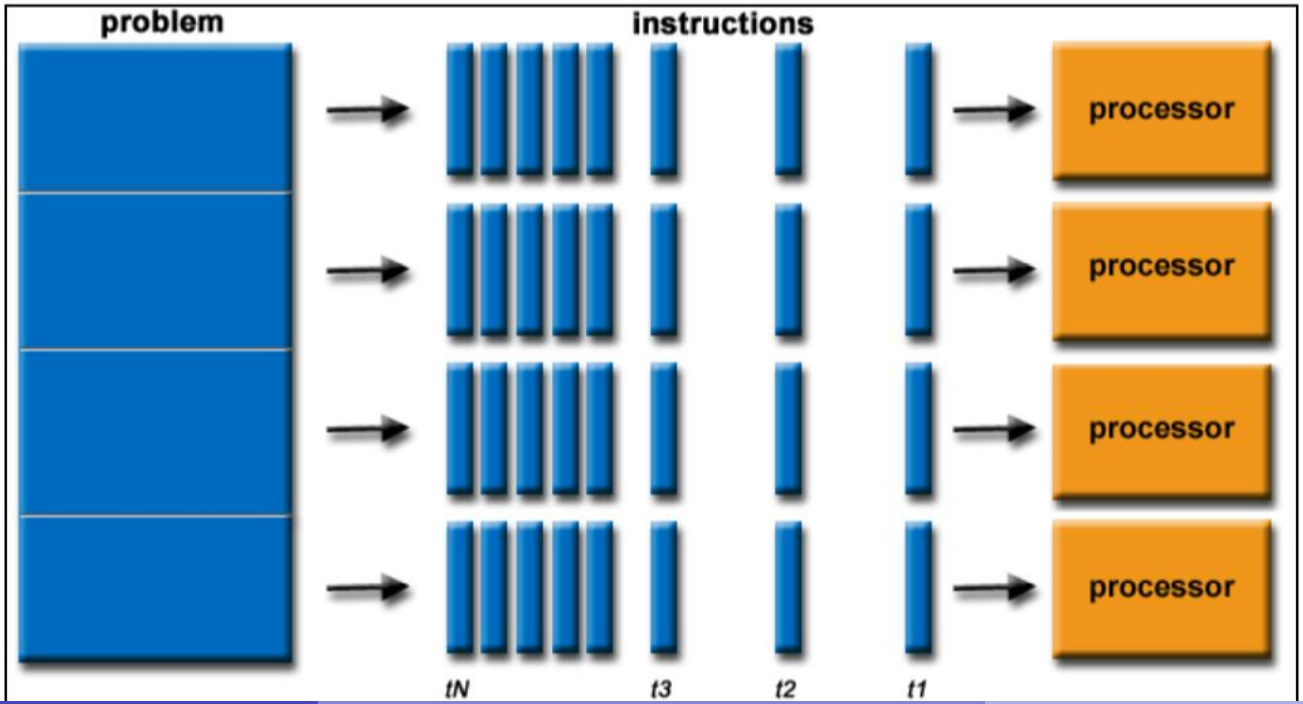
- 问题被分为一系列独立指令
- 指令按照先后顺序逐一执行
- 一般只在一个处理器上执行
- 在任何时间只有一条指令执行



什么是并行计算?

并行计算

- 问题被分为能够并发执行部分
- 每个部分进一步被分为一系列指令
- 来自不同部分的指令可以同时在不同处理器上执行
- 需要全局协同机制



为什么并行计算

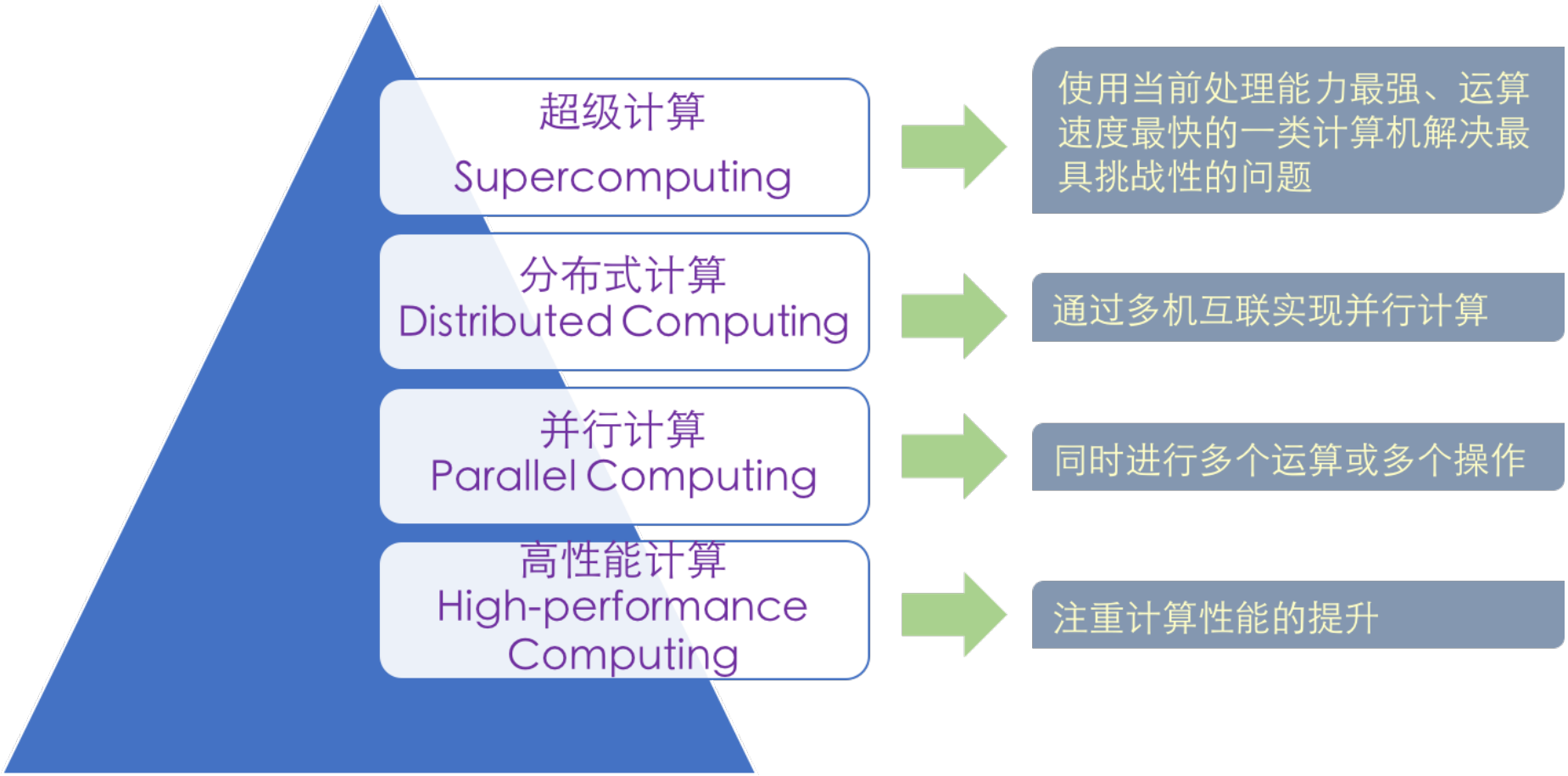
加速求解问题的速度

给定某应用，在单处理器上，串行执行需要 2 个星期（14 天），这个速度对一般的应用而言，是无法忍受的。而借助并行计算，使用 100 台处理器，如果加速 50 倍，将执行时间缩短为 6.72 个小时。

提高求解问题的规模

在单处理器上，受内存资源的限制，只能求解 10 万个未知数，但是当前数值模拟要求计算千万个未知数。于是，也可以借助并行计算，使用 100 个处理器，将问题求解规模线性地扩大 100 倍。

高性能计算、并行计算、分布式计算以及超级计算



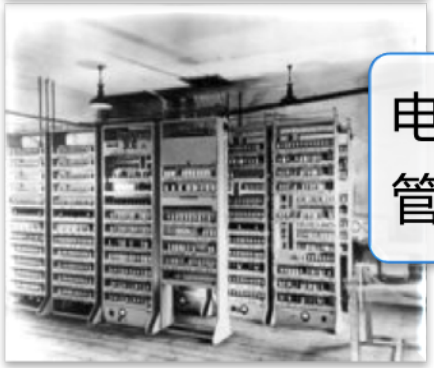
计算能力与存储能力的度量

前缀	简称	量级	计算能力	存储能力
Kilo-	K	10^3	KiloFLOPS (KFLOPS)	KiloByte (KB)
Mega-	M	10^6	MegaFLOPS (MFLOPS)	MegaByte (MB)
Giga-	G	10^9	GigaFLOPS (GFLOPS)	GigaByte (GB)
Tera-	T	10^{12}	TeraFLOPS (TFLOPS)	TeraByte (TB)
Peta-	P	10^{15}	PetaFLOPS (PFLOPS)	PetaByte (PB)
Exa-	E	10^{18}	ExaFLOPS (EFLOPS)	ExaByte (EB)
Zetta-	Z	10^{21}	ZettaFLOPS (ZFLOPS)	ZettaByte (ZB)
Yotta-	Y	10^{24}	YottaFLOPS (YFLOPS)	YottaByte (YB)

其中，FLOPS (flops or flop/s) 指每秒浮点运算次数：floating point operations per second.

计算机的发展历史

1946–1958



电子管

1958–1964

晶体管

1964–1972

集成电路

1972–今

VLSI

未来

量子？
生物？
光子？
...

萌芽：真空管电子计算机

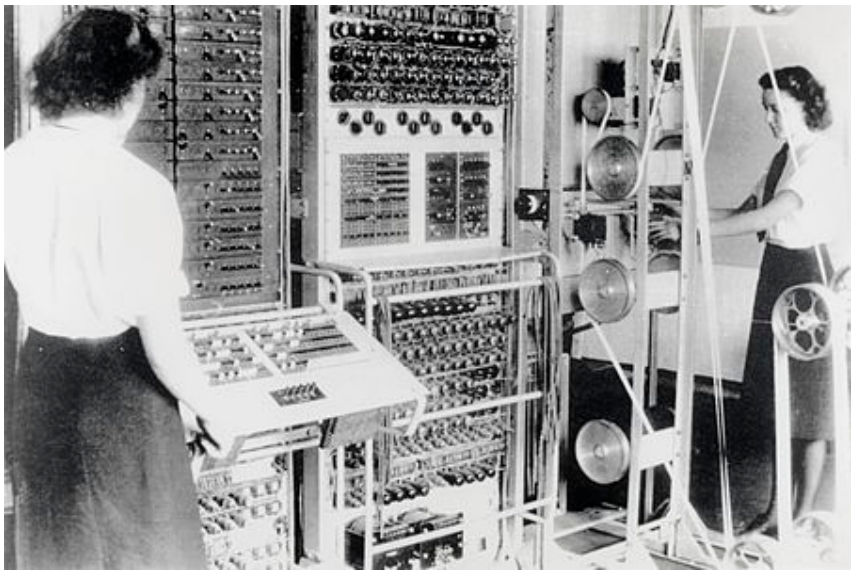
研发主要受第二次世界大战和冷战的军事需求推动。

Colossus

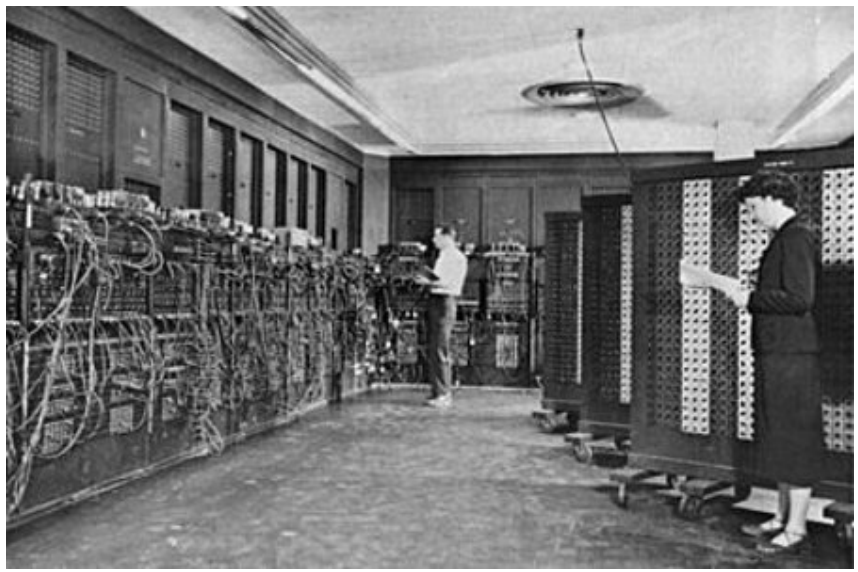
英国研制它来破解截获纳粹德国的无线电报信息。

ENIAC

美国研制它来快速计算炮兵射击表。



图：Colossus，1943，英国



图：ENIAC，1945，美国

早期：向量机

向量机（Vector Machine）使用单个指令同时处理多份数据。

Cray-1 向量机：

处理器个数	处理器频率	内存大小	存储大小	性能
1	80MHz	8.39 MB	303MB	160 MFLOPS

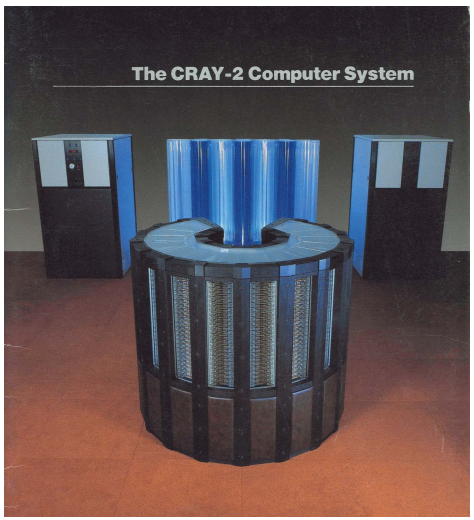
图：Cray-1，1976

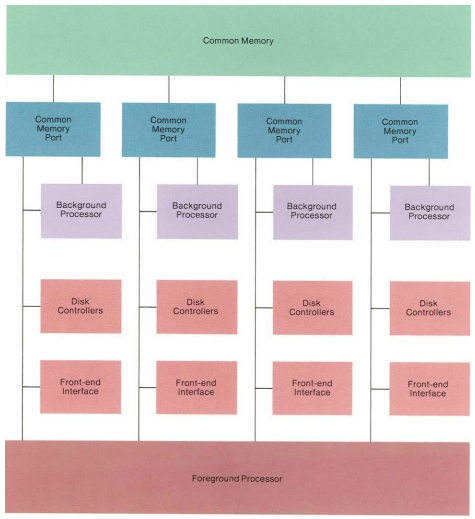
中期：并行向量处理机

并行向量处理机（Parallel Vector Processors，PVP）包括多个向量机，并通过共享内存实现交互。

Cray-2 多处理机：

处理器个数	处理器频率	内存大小	总性能
4	244MHz	256 MB	1.9 GFLOPS





图：Cray-2，1985

中后期：分布式并行机

分布式并行机（Parallel Processors，PP）通过高性能网络连接多个分布式存储节点，每个节点由商用微处理芯片组成。

Intel Paragon XP/S 140 并行机：

处理器个数	处理器性能	处理器频率	内存大小	访存带宽	网络带宽	总性能
3680	75 MFLOPS	50 MHz	128 MB	400 MB/s	175 MB/s	143 GFLOPS

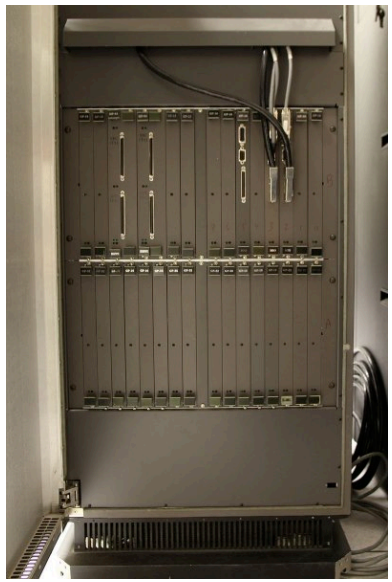


图: Intel Paragon XP/S 140，1994

中后期：对称多处理机

对称多处理机（Symmetric Multiprocessors，SMP）通过高性能网络连接多个高性能微处理芯片，芯片之间通过共享内存交互。

SUN Ultra E10000 多处理机：

处理器个数	处理器性能	处理器频率	内存大小	网络带宽	总性能
64	1 GFLOPS	250 MHz	64GB	12.8 GB/s	25 GFLOPS

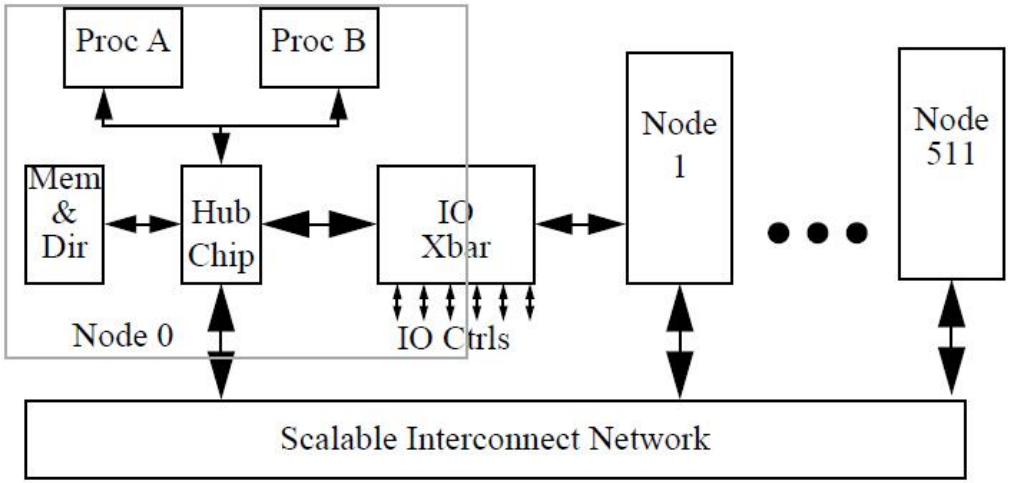
图：SUN Ultra E10000，1997

中后期：分布式共享并行机

分布式共享并行机（Distributed Share Memory，DSM）通过高性能网络连接多个高性能微处理芯片，每个芯片拥有局部内存，但所有局部内存都能实现全局共享。

SGI Origin 2000 并行机：

处理器个数	处理器频率	内存大小	网络带宽	总性能
128	195 MHz	512GB	1.5 GB/s	50 GFLOPS



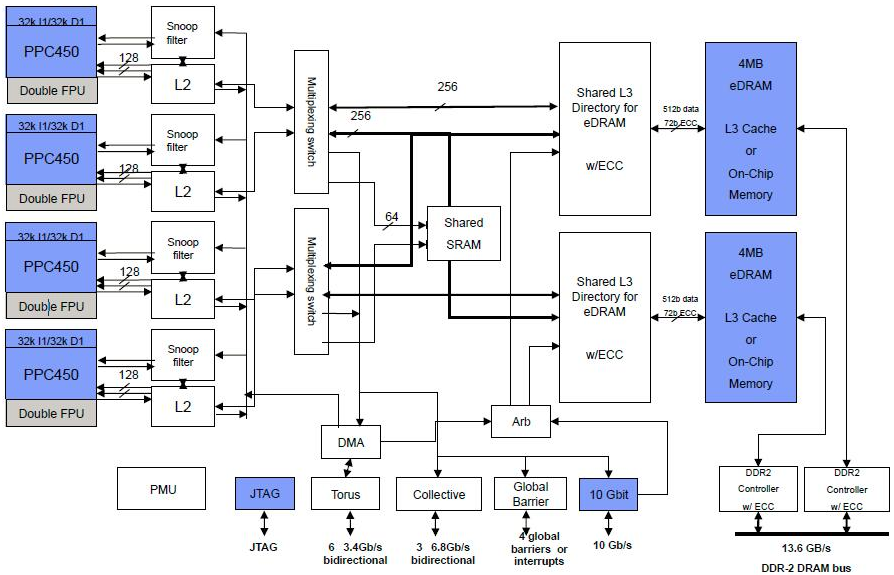
图：SGI Origin 2000，2002

近期：大规模并行机

大规模并行机（Massively Parallel Processors, MPP）通过高性能网络连接上万个分布式存储节点，每个节点包含多个高性能芯片。

Intrepid(Blue Gene/P) 并行机：

节点个数	节点核数	节点性能	处理器频率	网络带宽	总内存	总性能
40960	4	13.6 GFLOPS	850MHz	88 GB/s	80 TB	557 TFLOPS



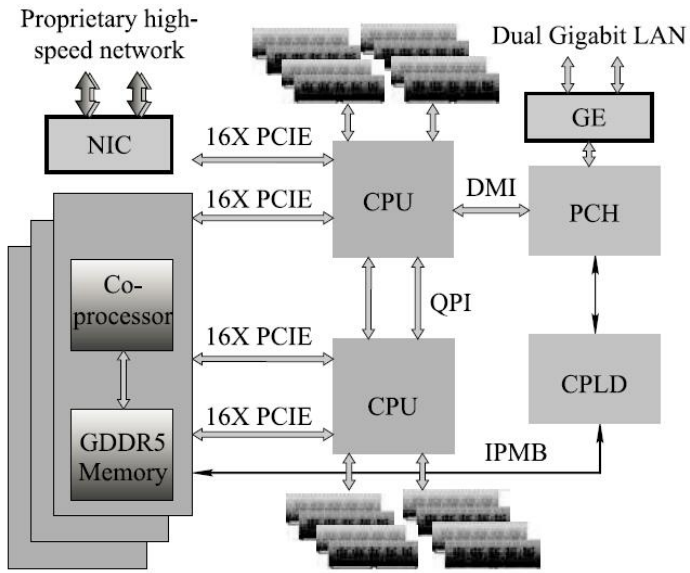
图：Intrepid(Blue Gene/P)，2008

近期：大规模加速并行机

大规模加速并行机（Massively Parallel Processors with Accelerators, MPPA）通过高性能网络连接上万个分布式存储节点，每个节点包含多个拥有加速器的高性能芯片。

天河 2 号：

节点个数	节点核数	节点性能	处理器频率	加速器	网络带宽	总内存	总性能
16,000	195	3.4 TFLOPS	2.2 GHz	3 × Xeon Phi	12 GB/s	1.34 PB	54.9 PFLOPS



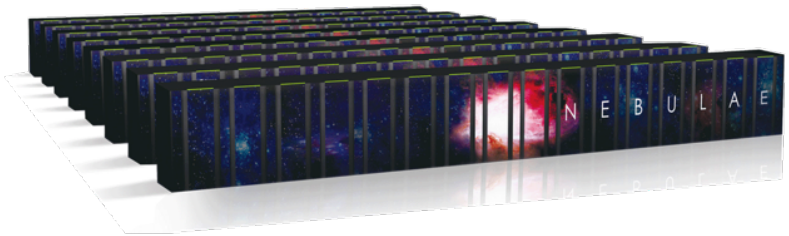
图：天河 2 号，2013

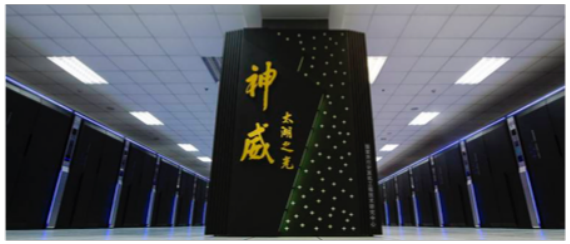
TOP500 排名

- TOP500 榜单每年更新两次，列举公开发布的最强性能的超级计算机前 500 名。
- 排榜根据 LINPACK 基准测试结果，程序核心是利用高斯消去法求解稠密线性系统。
- 网址：https://www.top500.org/

国产计算机 TOP500 入榜情况

- 联想深腾系列
  - 2003年11月，深腾 6800 Top500 第14位
  - 2009年6月，深腾7000 Top500 第19位
- 曙光系列
  - 2004年6月，曙光4000A Top500 第10位
  - 2009年6月，曙光5000A（魔方）Top500 第10位
  - 2010年11月，曙光6000（星云）Top500 第2位
- 天河（银河）系列
  - 2010年11月，天河1A Top500 第1位
  - 2013年6月、11月，天河2 Top500 第1位
  - 2014年6月、11月，天河2 Top500 第1位
  - 2015年6月、11月，天河2 Top500 第1位
- 神威（神州）系列
  - 2016年6月、11月，神威太湖之光 Top500 第1位
  - 2017年6月、11月，神威太湖之光 Top500 第1位





最新 TOP500 排名

#	Site	Manufacturer	Computer	Country	Cores	Rmax [Pflops]	Power [MW]
1	Oak Ridge National Laboratory	IBM	Summit IBM Power System, P9 22C 3.07GHz, Mellanox EDR, NVIDIA GV100	USA	2,414,592	148.6	10.1
2	Lawrence Livermore National Laboratory	IBM	Sierra IBM Power System, P9 22C 3.1GHz, Mellanox EDR, NVIDIA GV100	USA	1,572,480	94.6	7.4
3	National Supercomputing Center in Wuxi	NRCPC	Sunway TaihuLight NRCPC Sunway SW26010, 260C 1.45GHz	China	10,649,600	93.0	15.4
4	National University of Defense Technology	NUDT	Tianhe-2A ANUDT TH-IVB-FEP, Xeon 12C 2.2GHz, Matrix-2000	China	4,981,760	61.4	18.5
5	Texas Advanced Computing Center / Univ. of Texas	Dell	Frontera Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox HDR	USA	448,448	23.5	
6	Swiss National Supercomputing Centre (CSCS)	Cray	Piz Daint Cray XC50, Xeon E5 12C 2.6GHz, Aries, NVIDIA Tesla P100	Switzerland	387,872	21.2	2.38
7	Los Alamos NL / Sandia NL	Cray	Trinity Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries	USA	979,072	20.2	7.58
8	National Institute of Advanced Industrial Science and Technology	Fujitsu	AI Bridging Cloud Infrastructure (ABCI) PRIMERGY CX2550 M4, Xeon Gold 20C 2.4GHz, IB-EDR, NVIDIA V100	Japan	391,680	19.9	1.65
9	Leibniz Rechenzentrum	Lenovo	SuperMUC-NG ThinkSystem SD530, Xeon Platinum 8174 24C 3.1GHz, Intel Omni-Path	Germany	305,856	19.5	
10	Lawrence Livermore National Laboratory	IBM	Lassen IBM Power System, P9 22C 3.1GHz, Mellanox EDR, NVIDIA Tesla V100	USA	288,288	18.2	

图: 2019 年 6 月 TOP500 排名

TOP500 发展趋势：性能

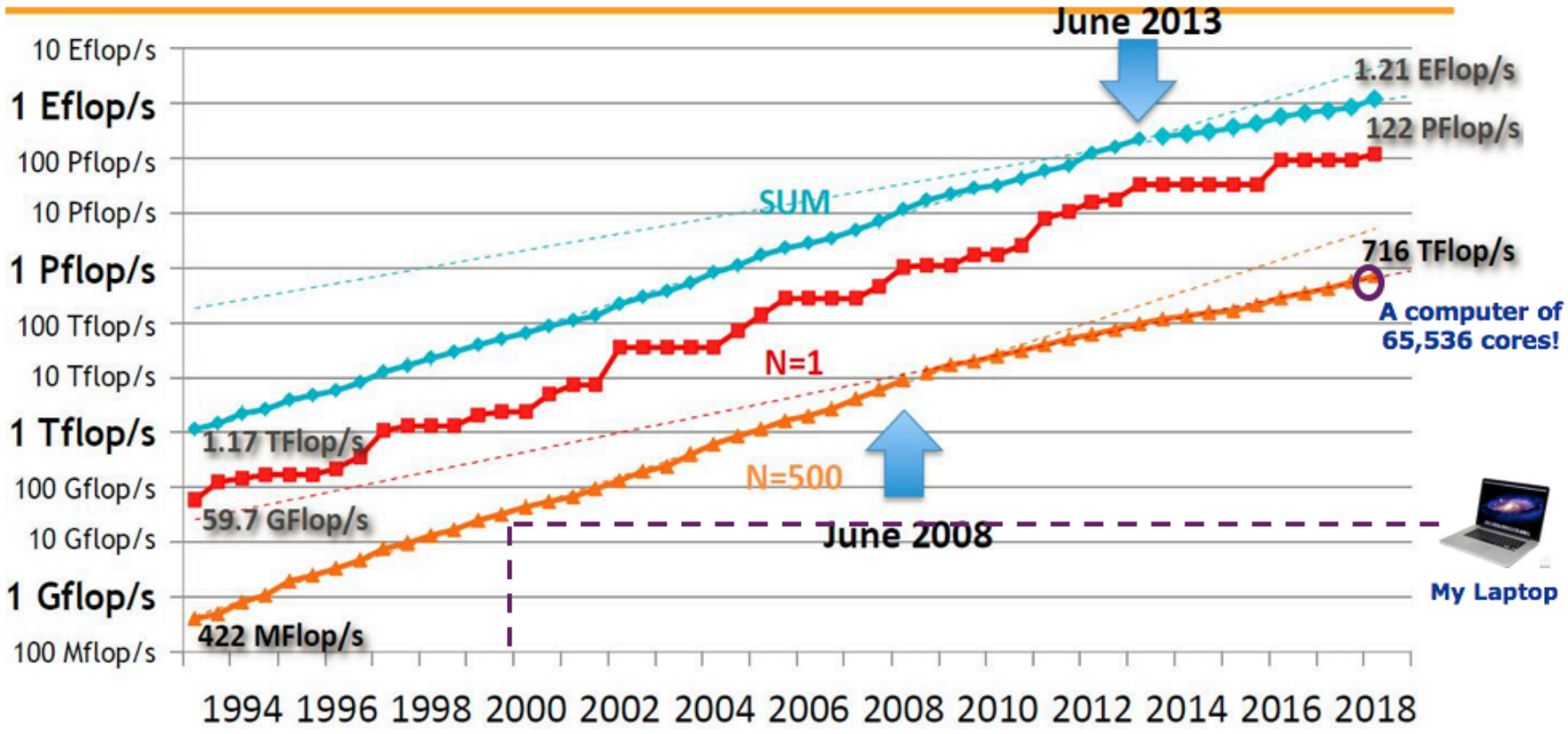


图: TOP500 性能发展趋势

TOP500 发展趋势：份额

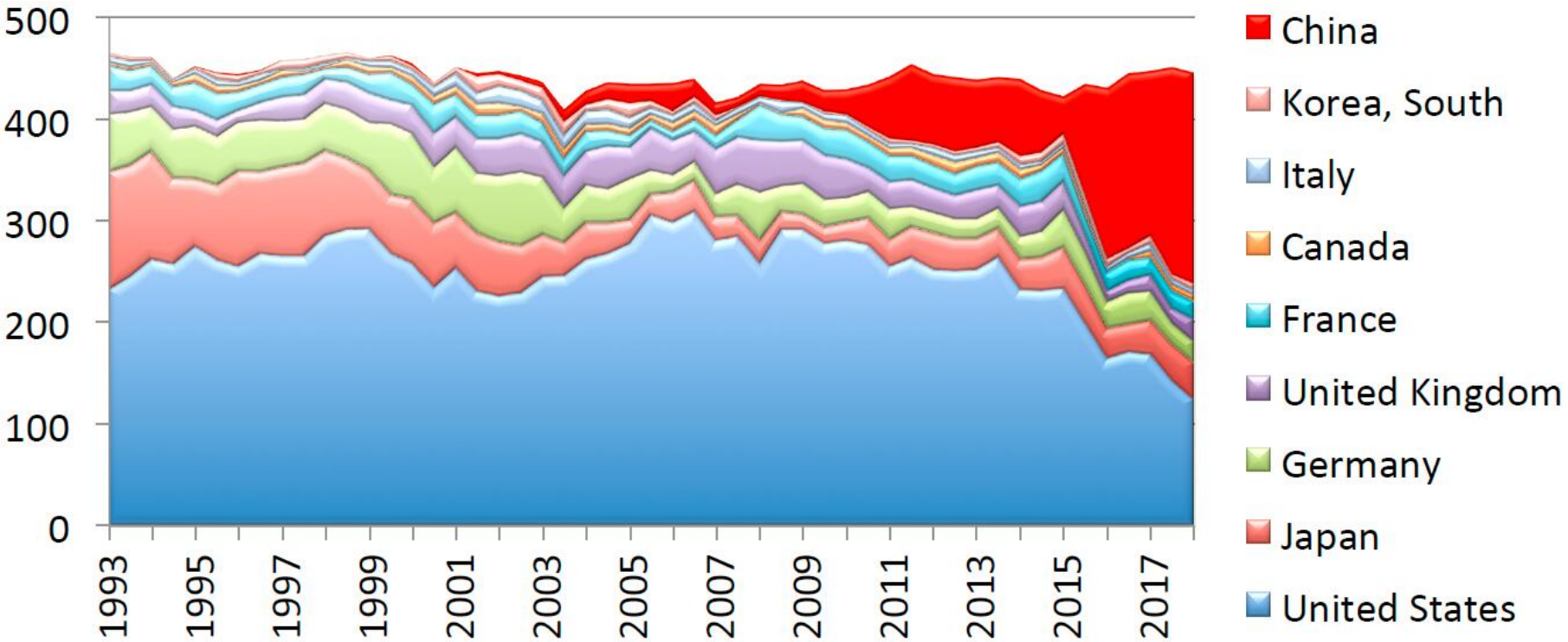
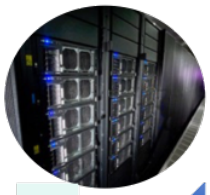


图: TOP500 中各国份额占比

TOP500 排名第一的超级计算机



美

2008-2010年
走鹃（美）
性能：1.0PF
结构：CPU-Cell
核数：12.96万

中

2010-2011年
天河一号A（中）
性能：2.5PF
结构：CPU-GPU
核数：18.6万

日

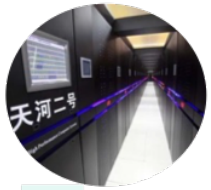
2011-2012年
京K（日）
性能：10.5PF
结构：CPU
核数：70.5万

美

2012年
红杉（美）
性能：17.2PF
结构：CPU
核数：157万

美

2012年
泰坦（美）
性能：17.6PF
结构：CPU-GPU
核数：56万



中

2013-2015年
天河二号（中）
性能：33.8PF
结构：CPU-MIC
核数：312万

中

2016年-2017
神威太湖之光（中）
性能：93PF
结构：SW
核数：1064万

美

2018
顶点（美）
性能：122PF
结构：CPU-GPU
核数：228万

硬件发展趋势

CPU-MIC异构计算

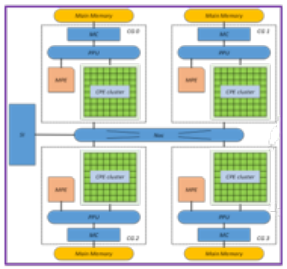
2个12核CPU
+
3块57核MIC

16000x

天河2（312万核）

片上异构计算

4个主核
+
256个从核



40000x

神威太湖之光（1064万核）

算法?
软件?

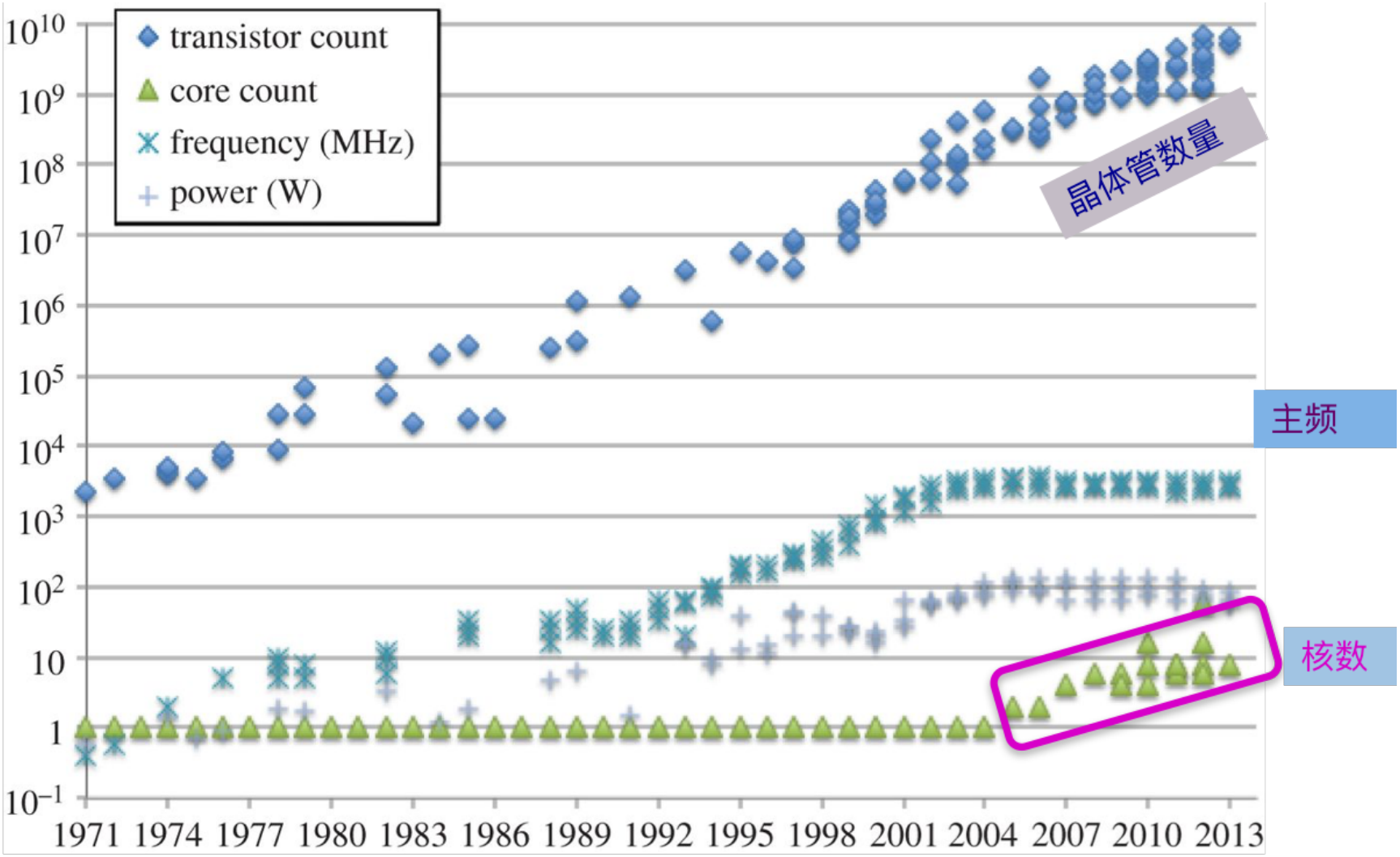
CPU-GPU异构计算

2个22核CPU
+
6块80核GPU

4608x

顶点（228万核）

为什么核数越来越多?

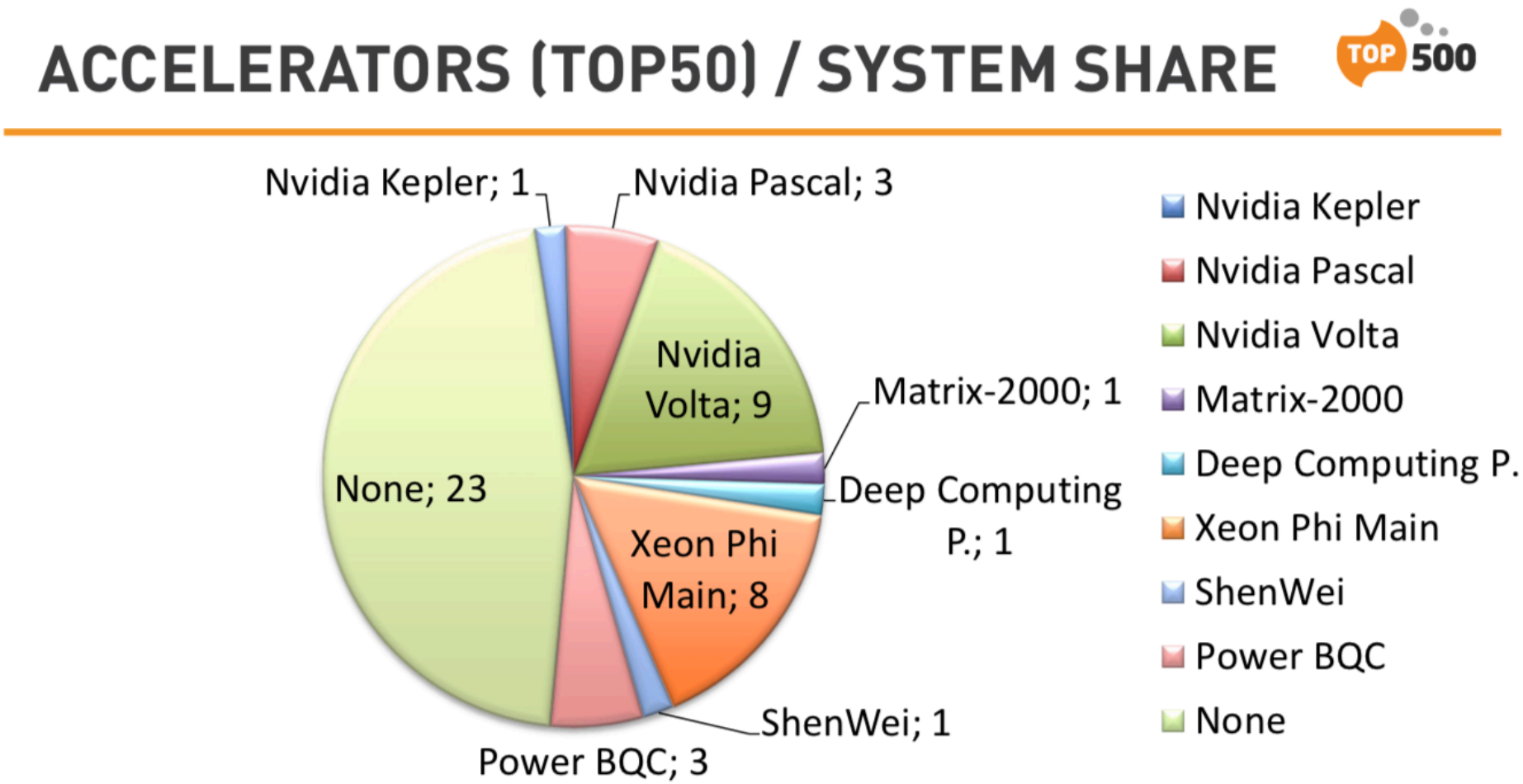


Courtesy: Giles & Reguly, 2014

为什么核数越来越多?

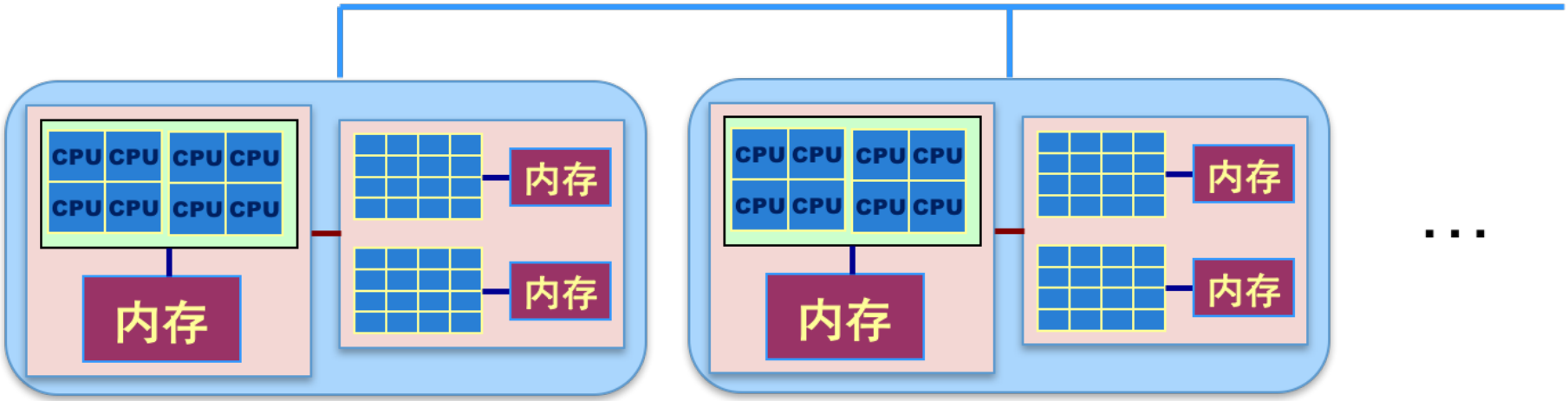
- 功耗问题成为了高性能计算机发展趋势变化的一大动因
  - 公式1：性能 $\propto$ 主频 x 核数
  - 公式2：功耗 $\propto$ 性能 x 电压^2
  - 公式3：主频 $\propto$ 电压
  - 提高性能方案1：提高主频
    - 为了提高性能1倍，主频增大1倍，但功耗提高7倍!
  - 提高性能方案2：提高核数
    - 为了提高性能1倍，核数增多1倍，功耗只需提高1倍!
    - 保持性能不变，核数增多1倍，功耗降低75%!
    - 保持功耗不变，核数增多1倍，性能提升58%!
- 结论：提高核数是维持性能提高并降低功耗增涨的有效途径!
  - 单核 → 双核 → 多核 → 众核 → ...

为什么采用异构设计?


ACCELERATORS (TOP50) / SYSTEM SHARE
TOP 500
Nvidia Kepler; 1
Nvidia Pascal; 3
Nvidia Volta; 9
Matrix-2000; 1
Deep Computing P.; 1
Xeon Phi Main; 8
ShenWei; 1
Power BQC; 3
None; 23
Nvidia Kepler
Nvidia Pascal
Nvidia Volta
Matrix-2000
Deep Computing P.
Xeon Phi Main
ShenWei
Power BQC
None

为什么采用异构设计?

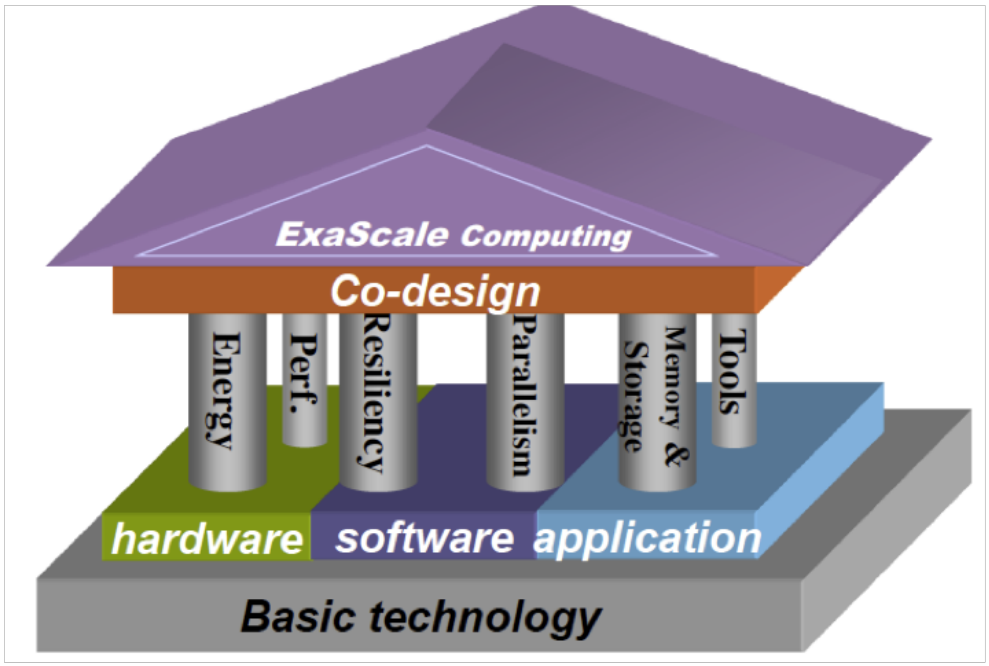
- 异构分布式并行机



- 异构：意味着处理器之间地位不平等，为什么?
  - 不同类型的处理器具备不同的能力
    - 处理复杂操作能力、计算能力、访存能力、通信能力等
  - 例子：CPU、GPU、MIC ...
  - 对比：军、师、旅、团、营、连、排...

并行算法与软件的研究价值

- 算法和软件是应用和计算机之间的桥梁
- 算法是软件的灵魂，不同类型的应用所需的算法可能不同
- 不同的算法各自适用于不同的计算机
  - 比如：传统的FFT算法在向量机上很好，但在分布式系统上不够理想
- 需要多方面专家协同设计（Co-design）

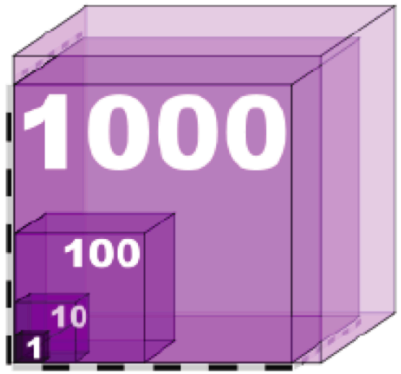


并行（与分布式）计算研究内容

- 硬件架构：认识当代高性能计算机体系架构特征，理解并行计算模型和并行性能评价方法，指导并行算法设计和并行程序实现。
- 并行算法：针对应用领域专家求解各类应用问题的计算方法，设计高效率的并行算法（将应用问题分解为可并行计算的多个子任务），并分析算法的可行性和效果。
- 并行编程：学习不同类型的高性能编程模型和工具，例如消息传递平台 MPI 或者共享存储平台 OpenMP，编程实现相应的并行算法，在此基础上结合高性能硬件特征和应用问题特性，优化程序性能。

并行计算关心的一些要点

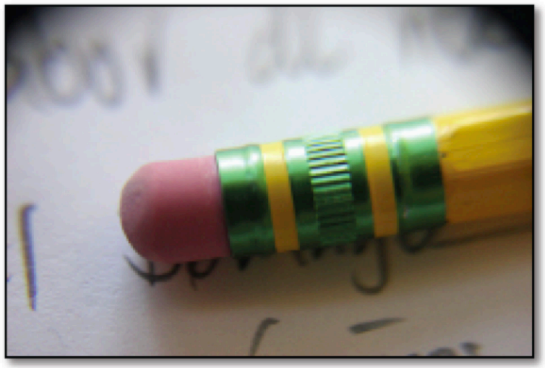
Time to Solution



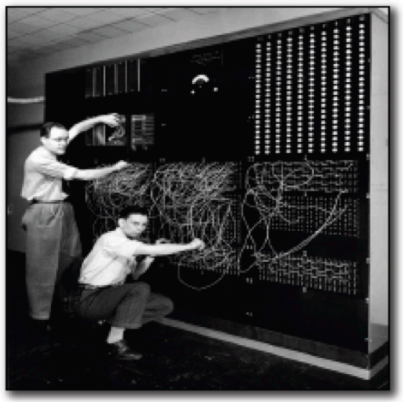
Scalability

Efficiency

Concurrency &
Data Locality



Resiliency



Programmability

参考文献 (1)


MPI Forum.
http://mpi-forum.org.


The OpenMP API specification for parallel programming.
http://openmp.org.


A Comprehensive MPI Tutorial Resource.
http://mpitutorial.com, 2018.


CUDA C Best Practices Guide, 2018.


CUDA C Programming Guide, 2018.


ACM Gordon Bell Prize.
https://awards.acm.org/bell.

参考文献 (2)

G. M. Amdahl.
Validity of the single processor approach to achieving large scale computing capabilities.
In *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, AFIPS '67 (Spring), pages 483–485, New York, NY, USA, 1967. ACM.

A. Baker.
COMP322: Fundamentals of Parallel Programming.
https://wiki.rice.edu/confluence/display/PARPROG/COMP322, 2018.

B. Barney.
EC3505: Message Passing Interface (MPI).
https://computing.llnl.gov/tutorials/mpi/, 2018.

参考文献 (3)


A. Barr.
CS 179: GPU Programming.
http://courses.cms.caltech.edu/cs179/, 2018.


R. Bendale, K. Jordan, J. Heyman, C. P. S. Brian, and S. B. Walkup.
Blue Gene/P Architecture, 2008.


Blaise Barney.
Introduction to parallel computing, 2018.
[Online; accessed 14-September-2018].


D. Culler, R. Karp, D. Patterson, A. Sahay, K. E. Schauser, E. Santos, R. Subramonian, and T. von Eicken.
LogP: Towards a realistic model of parallel computation.
In *Proceedings of the Fourth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '93, pages 1–12, New York, NY, USA, 1993. ACM.

参考文献 (4)

J. Demmel.
CS267: Applications of Parallel Computers.
https://people.eecs.berkeley.edu/~demmel/cs267_Spr16/, 2016.

V. Eijkhout.
Introduction to High Performance Scientific Computing.
lulu.com, 1st edition, 2015.

Encyclopedia of Parallel Computing.
https://link.springer.com/referencework/10.1007/978-0-387-09766-4, 2011.

参考文献 (5)

S. Fortune and J. Wyllie.
Parallelism in Random Access Machines.
In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA*, pages 114–118, 1978.

H. Fu, J. Liao, J. Yang, L. Wang, Z. Song, X. Huang, C. Yang, W. Xue, F. Liu, F. Qiao, W. Zhao, X. Yin, C. Hou, C. Zhang, W. Ge, J. Zhang, Y. Wang, C. Zhou, and G. Yang.
The Sunway TaihuLight Supercomputer: System and Applications.
pages 1–16.

F. Gebali.
Algorithms and Parallel Computing, volume 84.
John Wiley & Sons, 2011.

参考文献 (6)

M. Giles.
Course on CUDA Programming on NVIDIA GPUs.
https://people.maths.ox.ac.uk/gilesm/cuda/, 2018.

M. B. Giles and I. Reguly.
Trends in high-performance computing for engineering calculations.
Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 372(2022), 2014.

W. D. Gropp.
CS598: Designing and Building Applications for Extreme Scale Systems.
http://wgropp.cs.illinois.edu/courses/cs598-s16/, 2016.

J. L. Gustafson.
Reevaluating Amdahl's law.
Commun. ACM, 31(5):532–533, May 1988.

参考文献 (7)


G. Hager and G. Wellein.
Introduction to High Performance Computing for Scientists and Engineers.
CRC Press, Inc., 1st edition.


J. Hennessy and D. Patterson.
Computer Architecture: A Quantitative Approach.
Morgan Kaufmann, 6th edition edition, 2017.


A.-P. Hynninen.
CME 213: Introduction to Parallel Computing Using MPI, openMP, and CUDA.
http://web.stanford.edu/class/cme213/lecture.html, 2017.


Karp Challenge.
http://www.netlib.org/benchmark/karp-challenge, 1985.

参考文献 (8)


D. B. Kirk and W.-m. W. Hwu.
Programming Massively Parallel Processors: A Hands-on Approach.
Morgan Kaufmann, 3rd edition.


V. Kumar, G. Karypis, A. Gupta, and A. Grama.
Introduction to Parallel Computing.
Pearson, 2nd edition edition, 2003.


J. Laudon and D. Lenoski.
System overview of the SGI Origin 200/2000 product line.
In *Proceedings IEEE COMPCON 97. Digest of Papers*, pages 150–156.


X. Liao, L. Xiao, C. Yang, and Y. Lu.
MilkyWay-2 Supercomputer: System and Application.
8:345–356.

参考文献 (9)


P. Luszczek.
COSC462: Parallel Programming.
http://www.icl.utk.edu/~luszczek/teaching/courses/fall2016/cosc462/, 2016.


J. Mellor-Crummey.
COMP 422/534: Parallel Computing.
https://www.clear.rice.edu/comp422/, 2018.


H. Meuer, E. Strohmaier, J. Dongarra, H. Simon, and M. Martin.
Top 500 supercomputer lists, 1993.
[Online; accessed 14-September-2018].


J. Rehman.
Difference between serial and parallel processing, 2017.
[Online; accessed 14-September-2018].

参考文献 (10)


E. Solomonik.
CS 598: Communication Cost Analysis of Algorithms.
http://solomon2.web.engr.illinois.edu/teaching/cs598_fall2016/index.html, 2016.


X.-H. Sun.
Concurrent Average Memory Access Time (C-AMAT).
http://www.cs.iit.edu/~scs/research/c-amat/c-amat.html, 2017.


The UITS Knowledge Management team.
Understanding measures of supercomputer performance and storage system capacity, 2018.
[Online; accessed 14-September-2018].

参考文献 (11)


L. G. Valiant.
A bridging model for parallel computation.
Communications of the ACM, 33(8):103–111, Aug. 1990.


R. van Engelen.
ISC5318/CIS5930: High Performance Computing and Scientific Computing.
http://www.cs.fsu.edu/~engelen/courses/HPC/, 2017.


Wikipedia contributors, 2011.
[Online; accessed 14-September-2018].


S. Williams, A. Waterman, and D. Patterson.
Roofline: An insightful visual performance model for multicore architectures.
Communications of the ACM, 52(4):65–76, Apr. 2009.

参考文献 (12)


Y. Yan.
CSCE569: Parallel Computing.
https://passlab.github.io/CSCE569/, 2018.


T. Yang.
CS 240A: Applied Parallel Computing.
http://www.cs.ucsb.edu/~tyang/class/240a17/, 2017.


张林波，迟学斌，莫则尧，李若.
并行计算导论.
清华大学出版社, 2006.


陈国良.
并行计算: 结构算法编程 (第 3 版).
高等教育出版社, 第 3 版 edition, 2011 年 6 月 1 日.