



中国科学院大学
University of Chinese Academy of Sciences

计算机科学导论

学科展望

徐志伟 zxu@ict.ac.cn

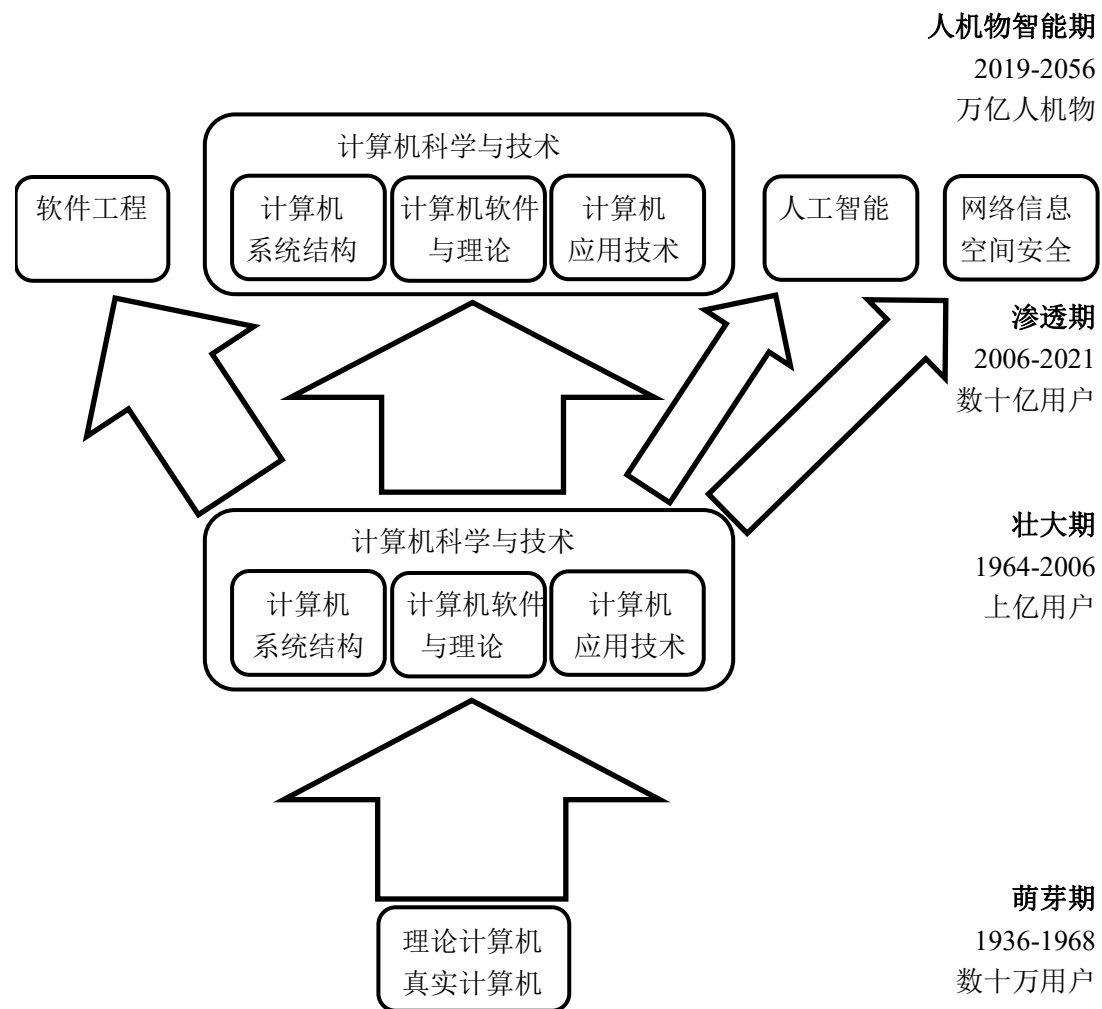
提纲

- 计算机学科展望
 1. 计算机学科发展的演化树
 2. 格雷12问题
 3. 科学研究的五个范式
 4. 21世纪文明需要创新型人才

课件中可能包含素材引用，特此致谢！

1. 计算机学科发展的演化树

- 1936-2056, 120年, 大体上可分为四个时期



计算机学科发展演化树

- 1936-2022，前三个时期

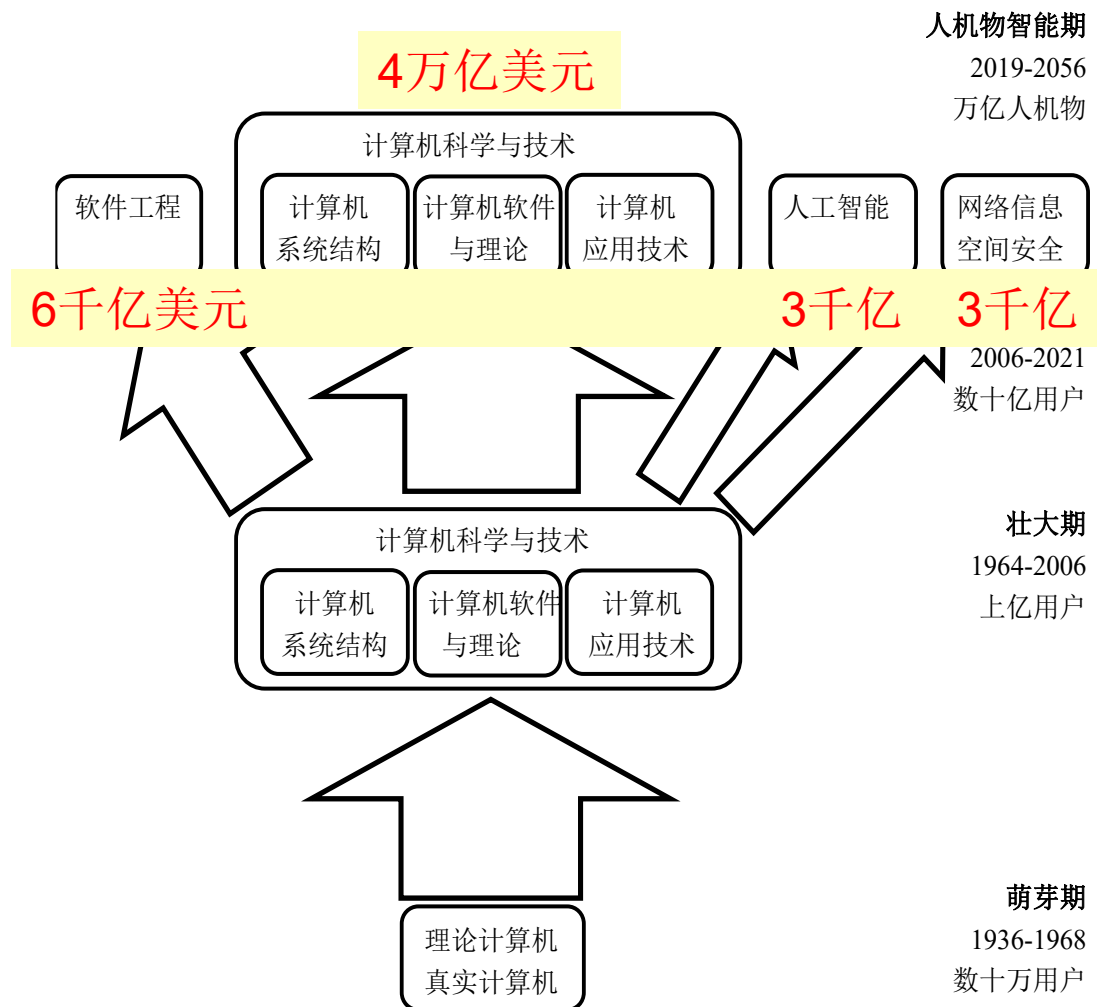
- 全球市场

2022: ~ 5.3 US\$ Trillions
5万亿美元
全球GDP=100万亿美元的5%

2000: ~ 1 US\$ Trillion
1万亿美元
全球GDP=34万亿美元的3%

1960: ~ 1 US\$ Billion, 十亿美元
全球GDP=1.4万亿美元的0.07%

1950: ~ US\$ Millions



计算机学科发展演化树

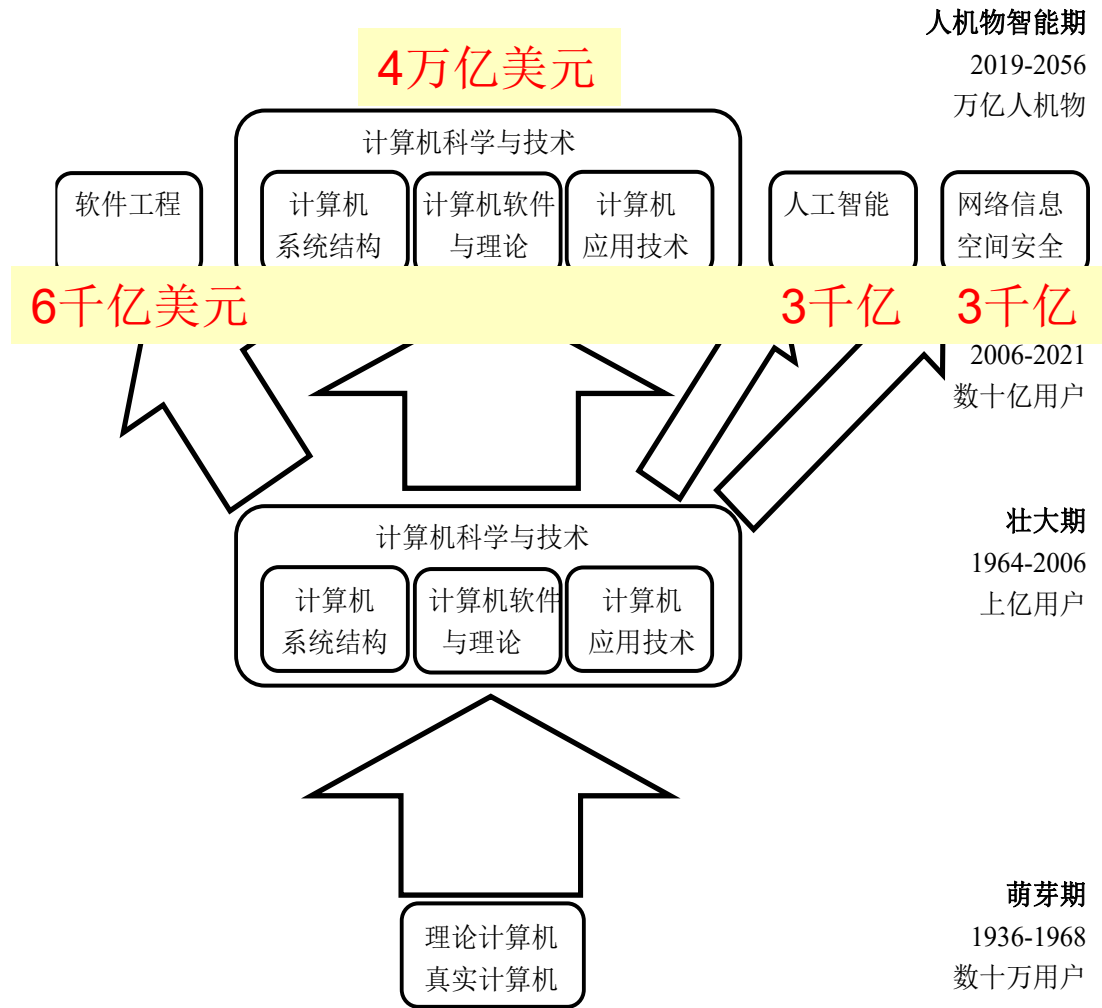
● 1936-2022，前三个时期

● 全球市场

- 2022: ~ 5.3 US\$ Trillions
5万亿美元
全球GDP=100万亿美元的5%
- 2000: ~ 1 US\$ Trillion
1万亿美元
全球GDP=34万亿美元的3%
- 1960: ~ 1 US\$ Billion，十亿美元
全球GDP=1.4万亿美元的0.07%
- 1950: ~ US\$ Millions

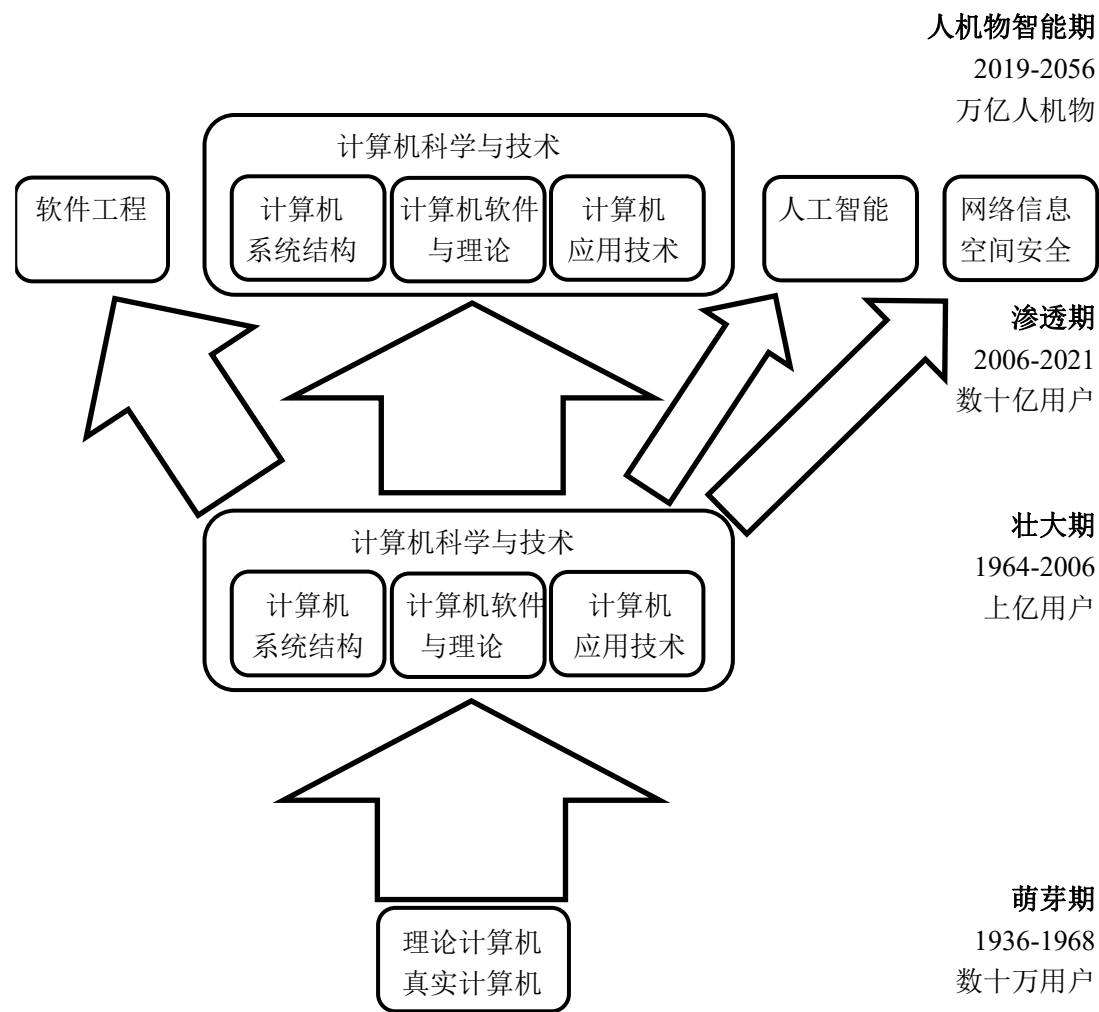
服务业提升
是大势所趋

产业占GDP比例	1992	2022
第三产业（服务业）	28%	53%
第二产业（工业）	42%	40%
第一产业（农业）	24%	7%



计算机学科发展演化树：萌芽期

- 1936：图灵机论文发表
- 1968：高德纳的《计算机程序设计艺术》教科书出版
 - 20世纪最有影响的12部科学专著之一
- 重要进展：物登明堂，裔裔皇皇
 - 通用的真实计算机出现
 - 冯诺依曼体系结构
 - 操作系统
 - 高级程序设计语言
 - 计算机应用涌现
 - 科学计算、计算机模拟仿真、计算机图形学、计算机过程控制、计算机信息管理系统等
 - 学科生态初步形成
 - 普度大学创立了全球第一个计算机专业
 - ACM、IEEE-CS、中国计算机学会成立



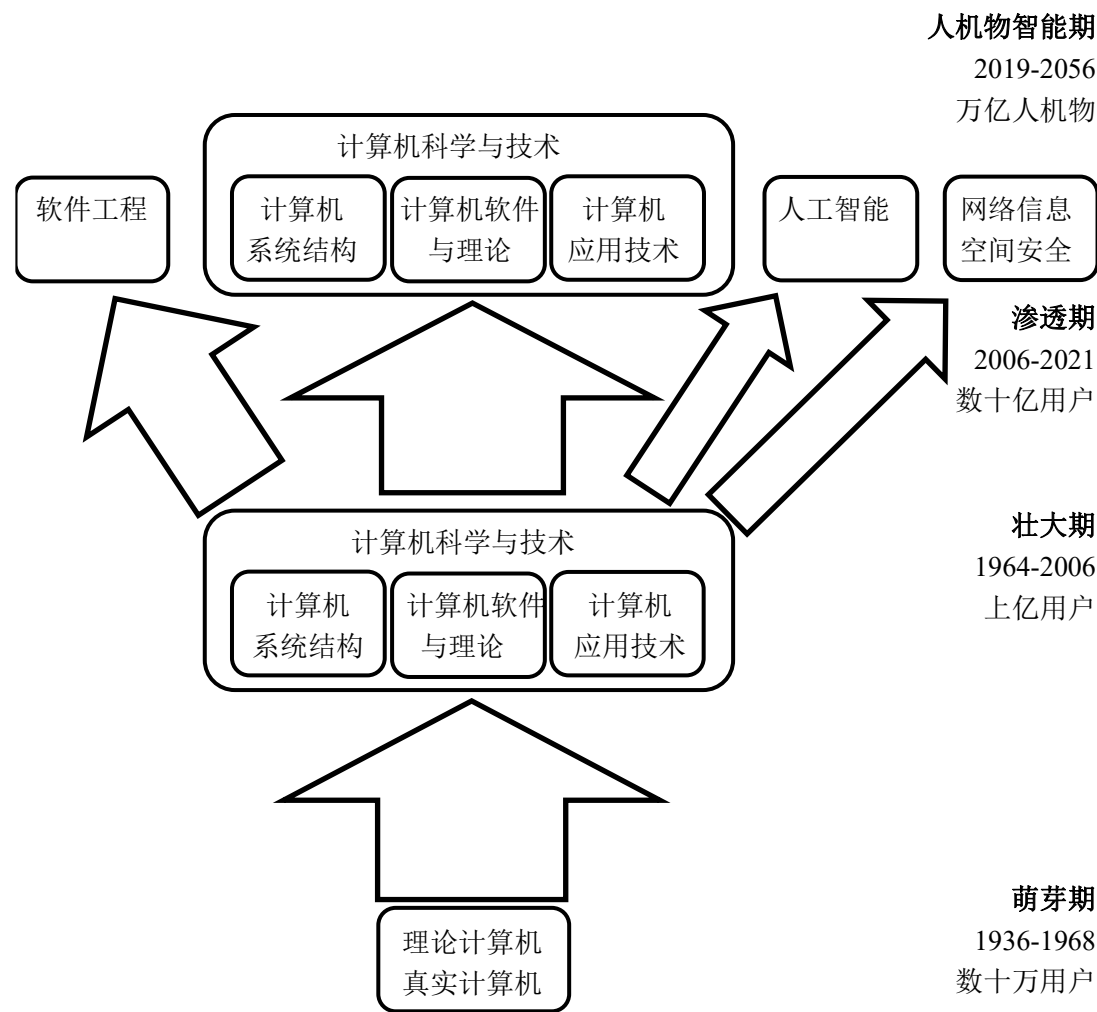
20世纪最有影响的12部科学专著

Philip, and Phylis Morrison. "100 or so BOOKS That Shaped a CENTURY of Science." American Scientist 87, no. 6 (1999): 542–53.

- 狄拉克：量子力学
- 爱因斯坦：瑞士论文集
- 曼德布罗特：分形
- 鲍林：化学键性质
- 罗素和怀特海德：数学基础
- 冯诺依曼和摩根斯坦：博弈论与经济行为
- 维纳：控制论
- 伍德沃和霍夫曼：轨域对称守恒
- 爱因斯坦：相对论的含义
- 费曼：量子电动力学
- 高德纳：计算机程序设计艺术
- Paul Dirac, Quantum Mechanics (1930)
- Albert Einstein, The Collected Papers of Albert Einstein: The Swiss Years: Writings, 1902–09 (1930)
- Benoit B. Mandelbrot, Fractals (1977)
- Linus Pauling, Nature of the Chemical Bond (1939)
- Bertrand Russell and Alfred North Whitehead, Principia Mathematica (1910–13, 3 vols.)
- Cyril Smith, Search For Structure (1981)
- John von Neumann and Oskar Morgenstern, Theory of Games and Economic Behavior (1944)
- Norbert Wiener, Cybernetics (1948)
- R. B. Woodward and Roald Hoffmann, Conservation of Orbital Symmetry (1970)
- Albert Einstein, The Meaning of Relativity (1922)
- Richard Feynman, QED (1985)
- Donald Knuth, The Art of Computer Programming (1968)

计算机学科发展演化树：壮大期

- 1964年：IBM发布S/360通用计算机系列
- 2006年：周以真发表“计算思维”论文
- 重要进展：以一耦万抽象栈成型
 - 计算局部性原理
 - P vs. NP，NP完备性
 - 分布式计算理论
- 计算机系统结构概念
 - 真实计算机**定性研究**
- 计算机体系结构：**量化研究**方法
 - 真实计算机定量研究
- 基础软件与应用软件
- 应对“软件危机”的“软件工程”
- 计算机网络
- 万维网



计算机学科发展演化树：壮大期

- 1964年：IBM发布S/360通用计算机系列
- 2006年：周以真发表“计算思维”论文
- 重要进展：以一耦万抽象栈成型
 - 计算局部性原理
 - P vs. NP, NP完备性
 - 分布式计算理论
 - 计算机系统结构概念
 - 真实计算机**定性研究**
 - 计算机体系结构：**量化研究**方法
 - 真实计算机定量研究
 - 基础软件与应用软件
 - 应对“软件危机”的“软件工程”
 - 计算机网络
 - 万维网

Hennessey & Patterson提出CPI公式

$$\frac{\text{Execution Time}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock Cycles}}{\text{Instruction}} \times \frac{\text{Clock Time}}{\text{Clock Cycle}}$$

程序执行时间 = 程序动态指令数 × CPI / 主频

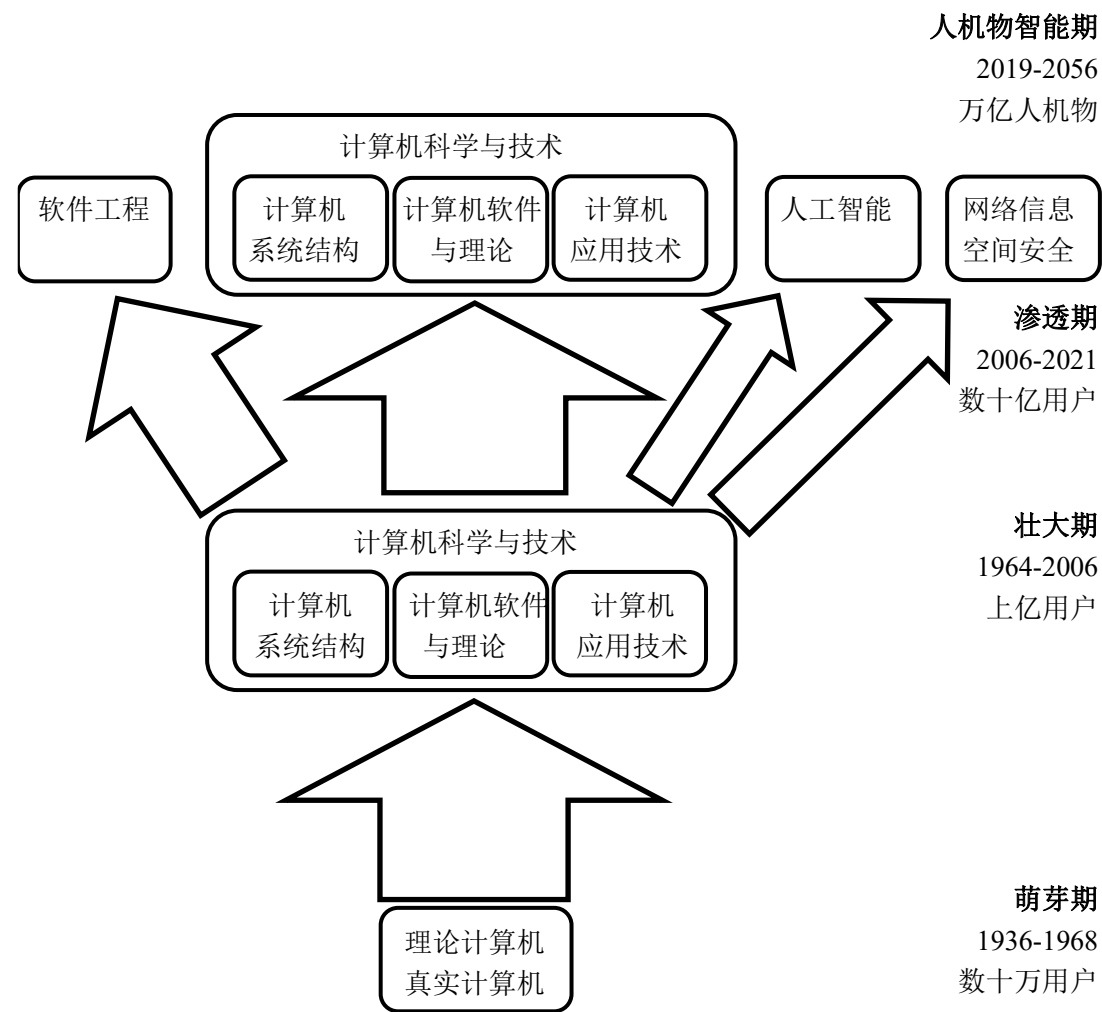
$$\text{处理器体系结构} = \begin{cases} \text{CISC 复杂指令集体系结构} & \text{CPI} > 1 \\ \text{RISC 精简指令集体系结构} & \text{CPI} = 1 \\ \text{Superscalar 超标量体系结构} & \text{CPI} < 1 \end{cases}$$

计算机学科发展演化树：渗透期

- 2006年：周以真发表“计算思维”论文
- 2021年：抖音成为全球访问量最大的网站，超过谷歌
 - 发展中国家也起到重要作用

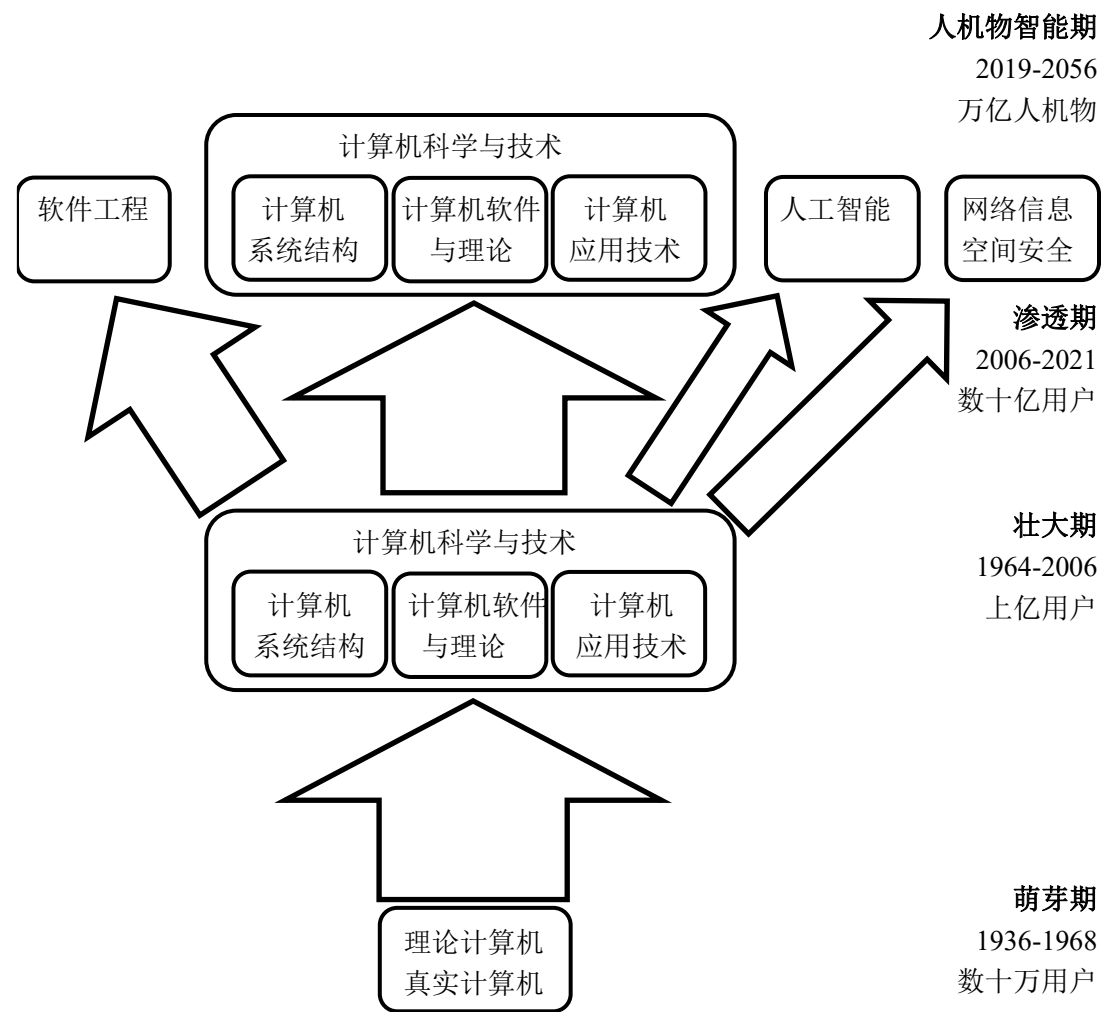
- 重要进展：计算机科学影响全球

- 计算思维愿景
- 移动互联网、云计算、大数据、人工智能等计算机技术迅猛发展，得到了大规模应用
- 渗透到了发展中国家
- 互联网全球渗透率
 - 2000: 6%
 - 2021: 59.5%
- 抖音（TikTok）成为2021年全球访问量最大的互联网网站，超过2020年冠军谷歌



计算机学科发展演化树：人机物智能期

- 2019年：业界提出**万亿设备新世界**愿景
(the Trillion-Device World)
 - 2018年**物端设备**数量超过**人端设备**
- 2056年：图灵机论文发表120周年
- 主要特征
 - 计算从赛博空间拓展到人机物三元计算世界
 - 计算机不只是电子计算机，它的部件还包括人和物
 - “人机物”融合的智能万物互联时代



2. 格雷12问题，对标数学的希尔伯特23问题

2050年前做到

- ① 可扩展系统（**Scalability**）：设计出算力可扩展1百万倍的系统结构。
- ② 图灵测试（**Turing Test**）：设计出可通过图灵测试的计算机系统。
- ③ 母语听（**Speech to text**）：构建计算机系统，能够像说母语的人一样听懂人讲话。
- ④ 母语说（**Text to speech**）：构建计算机系统，能够像说母语的人一样讲话。
- ⑤ 真人看（**See as well as a person**）。构建计算机系统，能够像人一样识别事物行为。
- ⑥ 个人数字资产库系统（**Personal Memex**）。记录个人一生所读、所看、所听，并能快速检索，但不做任何分析。消费者个人能够负担其购买成本与使用成本。
- ⑦ 全球数字资产库系统（**World Memex**）。所有文本、声音、图像、视频上网；专家级的分析和摘要能力；快速检索能力。
- ⑧ 远程呈现（**TelePresence**）。模拟人出现在另一个地方，能与当地环境和其他人交互，好像真实出现一样。
- ⑨ 无故障系统（**Trouble-Free Systems**）。构建计算机系统，能够供百万人日常使用，但只需一个兼职人员管理维护。
- ⑩ 安全系统（**Secure System**）。保障上述无故障系统只对授权用户开放、非法用户不能阻碍合法用户使用、信息不可能被窃取。证明这三种安全性。
- ⑪ 系统可用性（**AlwaysUp**）。保障上述无故障系统的可用性高达99.9999999%（9个9），即每100年才出错1秒。证明这种可用性。
- ⑫ 自动程序员（**Automatic Programmer**）。设计出一种规约语言或用户界面，具备5个性质：（1）通用，能够表达任意应用的设计；（2）高效，表达效率提升一千倍；（3）自动，计算机能够自动编译该设计表达；（4）易用，系统较为易用；（5）智能，可针对应用设计中存在的异常与缺失自动做推理、询问用户。

格雷12问题：性能

- ① **可扩展系统（Scalability）**：设计出算力可扩展**1百万倍**的系统结构。
- ② 图灵测试（Turing Test）：设计出可通过图灵测试的计算机系统。
- ③ 母语听（Speech to text）：构建计算机系统，能够像说母语的人一样听懂人讲话。
- ④ 母语说（Text to speech）：构建计算机系统，能够像说母语的人一样讲话。
- ⑤ 真人看（See as well as a person）。构建计算机系统，能够像人一样识别事物行为。
- ⑥ 个人数字资产库系统（Personal Memex）。记录个人一生所读、所看、所听，并能快速检索，但不做任何分析。消费者个人能够负担其购买成本与使用成本。
- ⑦ 全球数字资产库系统（World Memex）。所有文本、声音、图像、视频上网；专家级的分析和摘要能力；快速检索能力。
- ⑧ 远程呈现（TelePresence）。模拟人出现在另一个地方，能与当地环境和其他人交互，好像真实出现一样。
- ⑨ 无故障系统（Trouble-Free Systems）。构建计算机系统，能够供百万人日常使用，但只需一个兼职人员管理维护。
- ⑩ 安全系统（Secure System）。保障上述无故障系统只对授权用户开放、非法用户不能阻碍合法用户使用、信息不可能被窃取。证明这三种安全性。
- ⑪ 系统可用性（AlwaysUp）。保障上述无故障系统的可用性高达**99.9999999%**（**9个9**），即每**100年**才出错**1秒**。证明这种可用性。
- ⑫ 自动程序员（Automatic Programmer）。设计出一种规约语言或用户界面，具备**5个**性质：（1）通用，能够表达任意应用的设计；（2）高效，表达效率提升一千倍；（3）自动，计算机能够自动编译该设计表达；（4）易用，系统较为易用；（5）智能，可针对应用设计中存在的异常与缺失自动做推理、询问用户。

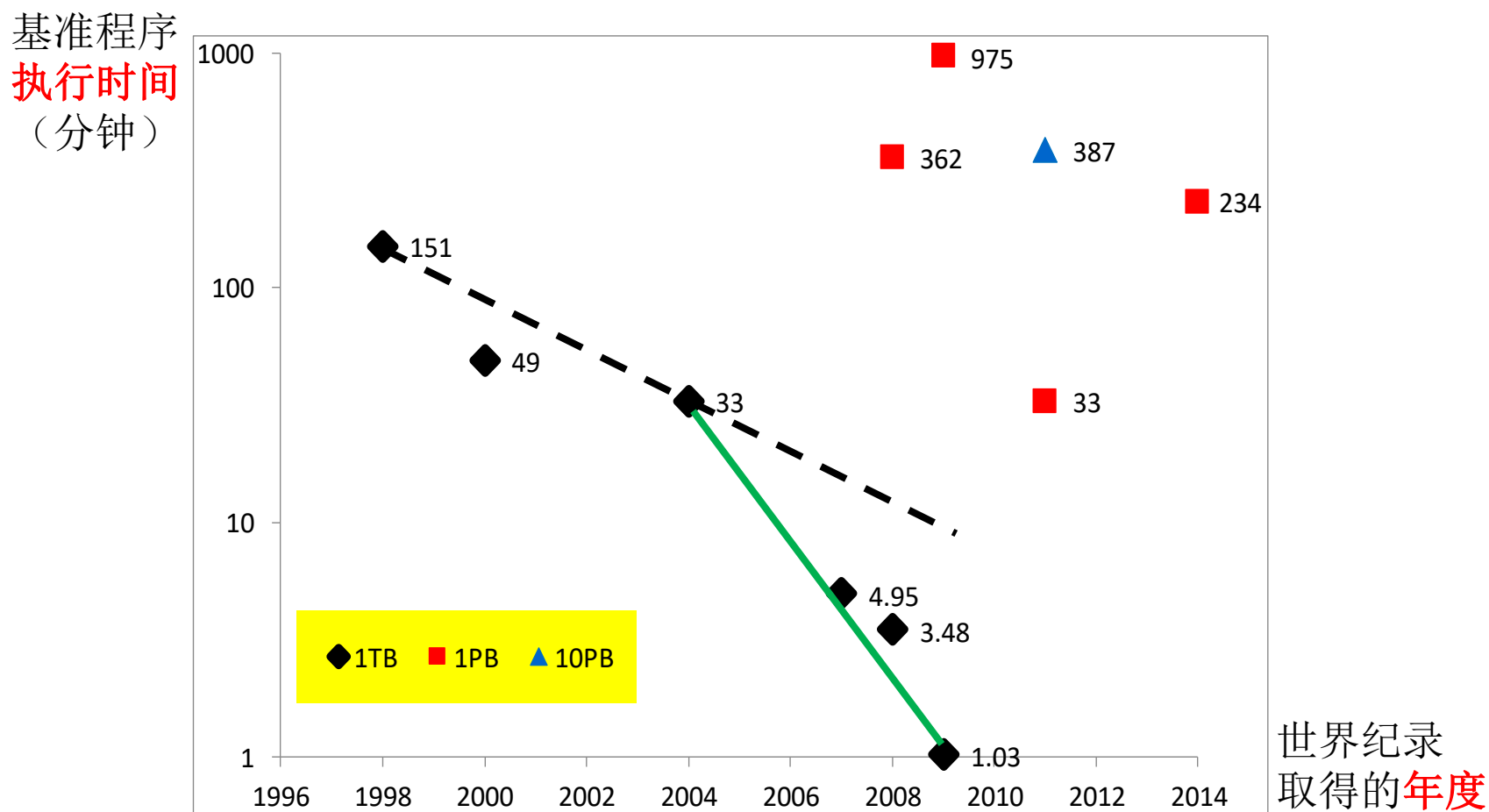
科学计算系统（超级计算机）已可扩展10万倍

- Linpack基准程序取得的实际计算速度最快，计算速度的单位是每秒执行的64比特浮点运算次数
- Fugaku采用了由多核并行计算节点连接而成的机群体系结构，整机系统结构可扩展，最多可添加到15万个计算节点，将计算速度提升到单节点计算速度的12万倍
 - Fugaku的“多核节点连接而成的机群”体系结构只能扩展12万倍

发布时间	1993年	2020年	1993-2020年增长倍数
冠军名称	Thinking Machine CM-5	Fujitsu Fugaku	N/A
问题规模	N = 52,224	N = 20,459,520	392
计算速度	59.7 GFlop/s	415,530 TFlop/s	6,960,302
主频	32 MHz	2.2 GHz	69
并发度	1,024 cores	7,299,072 cores	7,128
内存容量	32 GB	4,866,048 GB	152,064
功耗	96.5 KW	28,334.5 KW	294
成本	US \$30 million	US \$1 billion	33

大数据计算系统已可扩展上千倍

- 使用TeraSort基准程序，衡量进步的标准是排序1 TB需要多少时间
 - 前期进展很慢，采用“**机群+大数据应用框架**”后进展迅速
 - 数千个节点，整机比单节点快上千倍
 - 腾讯系统保持2016年的TeraSort世界纪录，每分钟排序60.7 TB

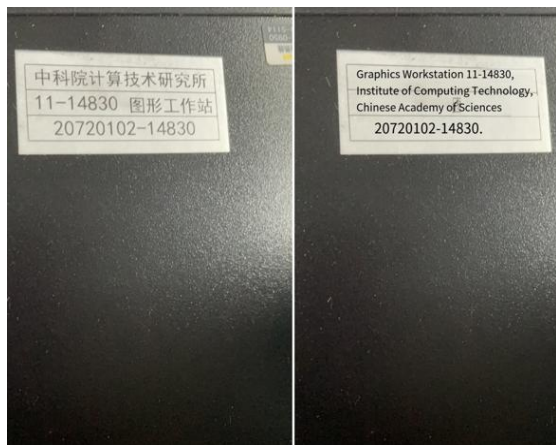


格雷12问题：智能

- ① 可扩展系统（**Scalability**）：设计出算力可扩展1百万倍的系统结构。
- ② 图灵测试（**Turing Test**）：设计出可通过图灵测试的计算机系统。
- ③ 母语听（**Speech to text**）：构建计算机系统，能够像说母语的人一样听懂人讲话。
- ④ 母语说（**Text to speech**）：构建计算机系统，能够像说母语的人一样讲话。
- ⑤ 真人看（**See as well as a person**）。构建计算机系统，能够像人一样识别事物行为。
- ⑥ 个人数字资产库系统（**Personal Memex**）。记录个人一生所读、所看、所听，并能快速检索，但不做任何分析。消费者个人能够负担其购买成本与使用成本。
- ⑦ 全球数字资产库系统（**World Memex**）。所有文本、声音、图像、视频上网；专家级的分析和摘要能力；快速检索能力。
- ⑧ 远程呈现（**TelePresence**）。模拟人出现在另一个地方，能与当地环境和其他人交互，好像真实出现一样。
- ⑨ 无故障系统（**Trouble-Free Systems**）。构建计算机系统，能够供百万人日常使用，但只需一个兼职人员管理维护。
- ⑩ 安全系统（**Secure System**）。保障上述无故障系统只对授权用户开放、非法用户不能阻碍合法用户使用、信息不可能被窃取。证明这三种安全性。
- ⑪ 系统可用性（**AlwaysUp**）。保障上述无故障系统的可用性高达99.9999999%（9个9），即每100年才出错1秒。证明这种可用性。
- ⑫ 自动程序员（**Automatic Programmer**）。设计出一种规约语言或用户界面，具备5个性质：（1）通用，能够表达任意应用的设计；（2）高效，表达效率提升一千倍；（3）自动，计算机能够自动编译该设计表达；（4）易用，系统较为易用；（5）智能，可针对应用设计中存在的异常与缺失自动做推理、询问用户。

智能应用已有明显进展

- ChatGPT, GPT-4, 深度求索 (DeepSeek)
- 进步主要得益于深度学习技术的发展和
 - 算法 + 算力 + 数据, 以及整合这些资源的系统
- 但离格雷的目标还差得远



徐志伟问: Turing machine has finite states. But random access machine seems to violate this finite state assumption. How to reconcile this contradiction?

.....

Reconciling the "Contradiction":

1. **Both have Finite Control:** The core similarity is that **both models have a finite control mechanism.**
 - TM: Finite set of states + fixed transition table.
 - RAM: Finite program (sequence of instructions).
 - This finite control governs the computation step-by-step.
2. **Both have Unbounded Storage:**
 - TM: Infinite tape.
 - RAM: Unbounded register array.
 - This unbounded storage is where data (and intermediate computational state like head position or memory addresses) resides and grows as needed.
3. **"State" in the Church-Turing Sense:** When we say a TM has "finite states," we specifically mean its *control unit* has a finite number of internal configurations. The *complete instantaneous description* (control state + head position + tape content) is *not* finite. Similarly, the RAM's *control flow* is determined by a finite program, while its *complete state* (PC + all memory contents) is *not* finite.

Conclusion:

There is no fundamental contradiction. The "finite states" property refers **only to the control mechanism** of both the Turing Machine *and* the Random Access Machine. Both models supplement this finite control with **unbounded external storage** (tape or memory registers). The ability of the RAM to directly access any memory location does not violate the finite state principle of computation; it simply provides a different, more efficient access paradigm to the unbounded storage resource that is also fundamentally present (though sequentially accessed) in the Turing Machine model. Both models capture the same notion of effective computation.

网易拍照翻译 (左)、讯飞翻译机 (中)、百度智能音箱 (右)

格雷12问题：品质

- ① 可扩展系统（Scalability）：设计出算力可扩展1百万倍的系统结构。
- ② 图灵测试（Turing Test）：设计出可通过图灵测试的计算机系统。
- ③ 母语听（Speech to text）：构建计算机系统，能够像说母语的人一样听懂人讲话。
- ④ 母语说（Text to speech）：构建计算机系统，能够像说母语的人一样讲话。
- ⑤ 真人看（See as well as a person）。构建计算机系统，能够像人一样识别事物行为。
- ⑥ 个人数字资产库系统（Personal Memex）。记录个人一生所读、所看、所听，并能快速检索，但不做任何分析。消费者个人能够负担其购买成本与使用成本。
- ⑦ 全球数字资产库系统（World Memex）。所有文本、声音、图像、视频上网；专家级的分析和摘要能力；快速检索能力。
- ⑧ 远程呈现（TelePresence）。模拟人出现在另一个地方，能与当地环境和其他人交互，好像真实出现一样。
- ⑨ 无故障系统（Trouble-Free Systems）。构建计算机系统，能够供百万人日常使用，但只需一个兼职人员管理维护。
- ⑩ 安全系统（Secure System）。保障上述无故障系统只对授权用户开放、非法用户不能阻碍合法用户使用、信息不可能被窃取。证明这三种安全性。
- ⑪ 系统可用性（AlwaysUp）。保障上述无故障系统的可用性高达99.9999999%（9个9），即每100年才出错1秒。证明这种可用性。
- ⑫ 自动程序员（Automatic Programmer）。设计出一种规约语言或用户界面，具备5个性质：（1）通用，能够表达任意应用的设计；（2）高效，表达效率提升一千倍；（3）自动，计算机能够自动编译该设计表达；（4）易用，系统较为易用；（5）智能，可针对应用设计中存在的异常与缺失自动做推理、询问用户。

云账户互联网服务系统

- 为6600万自由职业者提供秒批办照、收入结算、税款代缴、保险保障服务

- 2016年成立，2021年实现收入500多亿元、纳税30多亿元
- 云账户董事长获“全国脱贫攻坚先进个人”荣誉
- 仅有200多名技术人员

- 得益于分布式系统进展，使用了30余种开源应用框架



- 格雷问题进步

- 第9问题：无故障系统，供百万人日常使用，只需一个兼职人员管理维护
 - 云账户系统支持6600多万活跃用户，需要三十余名全职工程师做运行维护

云账户互联网服务系统

- 为6600万自由职业者提供秒批办照、收入结算、税款代缴、保险保障服务

- 2016年成立，2021年实现收入500多亿元、纳税30多亿元；
- 云账户董事长获“全国脱贫攻坚先进个人”荣誉
- 仅有200多名技术人员

- 得益于分布式系统进展，使用了30余种开源应用框架



- 格雷问题进步

- 第10问题：安全系统。保障上述无故障系统只对授权用户开放、非法用户不能阻碍合法用户使用、信息不可能被窃取。证明这三种安全性。
 - 云账户系统实际运营体现了上述三种安全性，但不能数学严密地证明

云账户互联网服务系统

- 为6600万自由职业者提供秒批办照、收入结算、税款代缴、保险保障服务

- 2016年成立，2021年实现收入500多亿元、纳税30多亿元；
- 云账户董事长获“全国脱贫攻坚先进个人”荣誉
- 仅有200多名技术人员

- 得益于分布式系统进展，使用了30余种开源应用框架



- 格雷问题进步

- 第11问题：可用性高达99.9999999%（9个9），每100年才出错1秒。证明这种可用性。
 - 云账户系统已经在五年多时间内无中断地为用户服务
 - 可用性还不能证明

格雷12问题：易用

- ① 可扩展系统（**Scalability**）：设计出算力可扩展1百万倍的系统结构。
- ② 图灵测试（**Turing Test**）：设计出可通过图灵测试的计算机系统。
- ③ 母语听（**Speech to text**）：构建计算机系统，能够像说母语的人一样听懂人讲话。
- ④ 母语说（**Text to speech**）：构建计算机系统，能够像说母语的人一样讲话。
- ⑤ 真人看（**See as well as a person**）。构建计算机系统，能够像人一样识别事物行为。
- ⑥ 个人数字资产库系统（**Personal Memex**）。记录个人一生所读、所看、所听，并能快速检索，但不做任何分析。消费者个人能够负担其购买成本与使用成本。
- ⑦ 全球数字资产库系统（**World Memex**）。所有文本、声音、图像、视频上网；专家级的分析和摘要能力；快速检索能力。
- ⑧ 远程呈现（**TelePresence**）。模拟人出现在另一个地方，能与当地环境和其他人交互，好像真实出现一样。
- ⑨ 无故障系统（**Trouble-Free Systems**）。构建计算机系统，能够供百万人日常使用，但只需一个兼职人员管理维护。
- ⑩ 安全系统（**Secure System**）。保障上述无故障系统只对授权用户开放、非法用户不能阻碍合法用户使用、信息不可能被窃取。证明这三种安全性。
- ⑪ 系统可用性（**AlwaysUp**）。保障上述无故障系统的可用性高达99.9999999%（9个9），即每100年才出错1秒。证明这种可用性。
- ⑫ **自动程序员（Automatic Programmer）**。设计出一种规约语言或用户界面，具备5个性质：（1）通用，能够表达任意应用的设计；（2）高效，表达效率提升一千倍；（3）自动，计算机能够自动编译该设计表达；（4）易用，系统较为易用；（5）智能，可针对应用设计中存在的异常与缺失自动做推理、询问用户。

云账户互联网服务系统

- 为6600万自由职业者提供秒批办照、收入结算、税款代缴、保险保障服务

- 2016年成立，2021年实现收入500多亿元、纳税30多亿元；
- 云账户董事长获“全国脱贫攻坚先进个人”荣誉
- 仅有200多名技术人员

- 得益于分布式系统进展，使用了30余种开源应用框架



- 格雷问题进步

- 第12问题：自动程序员。设计出一种规约语言或用户界面，使得表达效率提升一千倍。
 - 云账户仍然主要依赖于产品经理和工程师的合作来设计、开发和测试系统
 - 虽然利用基础组件库、微服务、开发工具链、敏捷开发流程等技术可减少重复工作，但云账户团队主要基于高级编程语言开发分布式应用系统
 - 只在几个限定的应用场景中，才支持最终用户在可视化环境中拖拽完成系统设计
 - 尚缺乏可自动执行的系统抽象

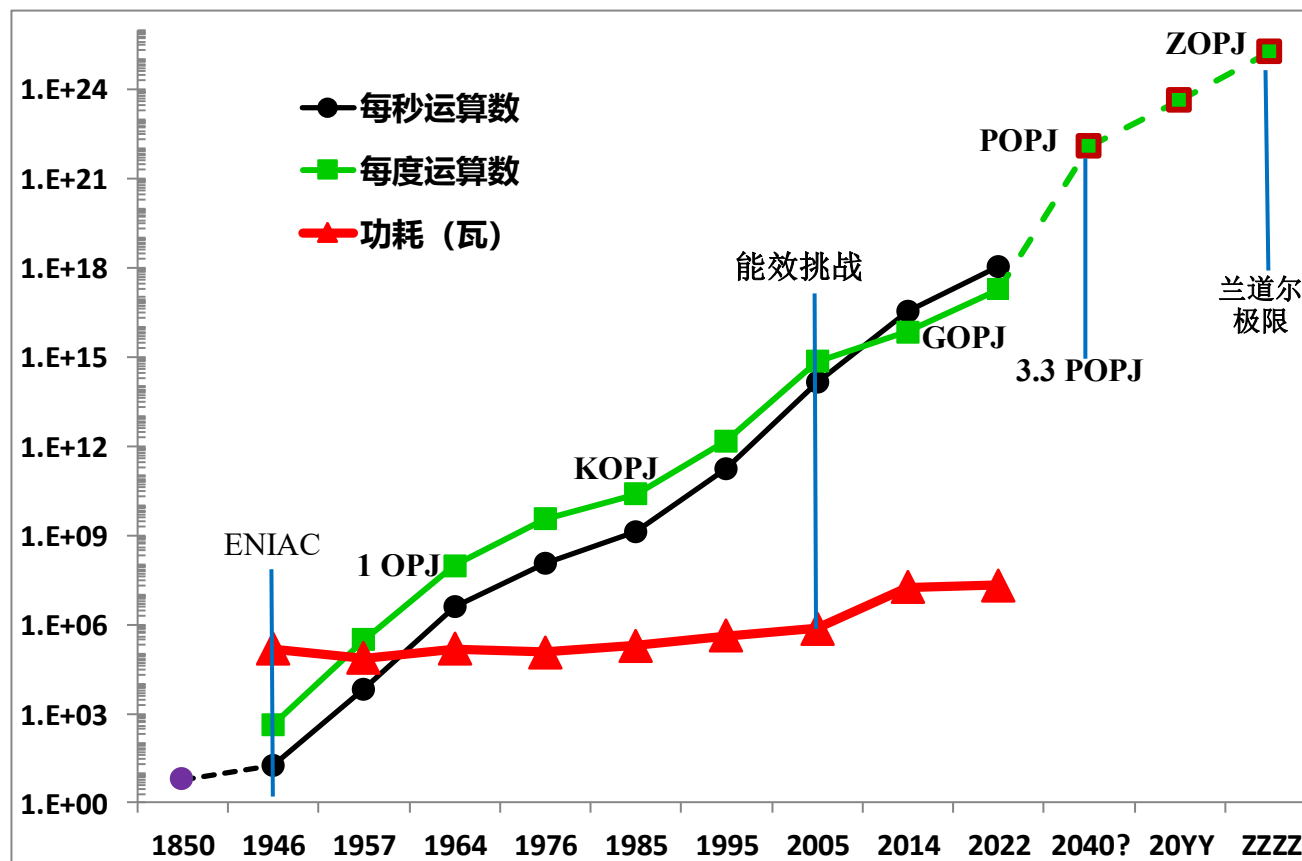
解决格雷问题受制于能效挑战

- OPJ: 每焦耳运算数
- 物理学的兰道尔极限前的三个里程碑
 - ENIAC: 终结了95年缓慢进展, 开启了60年的指数改善
 - 2005年开始, 能效挑战变得严峻
 - DARPA提出3.3 POPJ的目标, 试图大幅提升能效

Z: Zetta, 泽, 10^{21}
P: Peta, 拍, 10^{15}
G: Giga, 吉, 10^9
K: Kilo, 千, 10^3

单套计算机系统的
速度、能效、功耗

发展历史与展望



计算思维展望

- 数字符号
 - 串行计算过程
 - 图灵可计算性
- ➔ 数字模拟系统（digital and analog system）
 - ➔ 并行分布式系统、全球算力网（信息高铁）
 - ➔ 图灵可计算性 支撑的 非图灵可计算性



3. 科学研究的第五范式：AI4R (AI for Research)

- AI4R的别称

- AI4S(AI for Science), AI4SS(AI for Social Science), AI4E(AI for Education)
 - AI for Math, AI for Physics, AI for Chemistry, AI for Biology, ...

- 科学研究的范式演变（paradigm shift）

- 理论，实验，科学计算（第三范式，第四范式，第五范式）

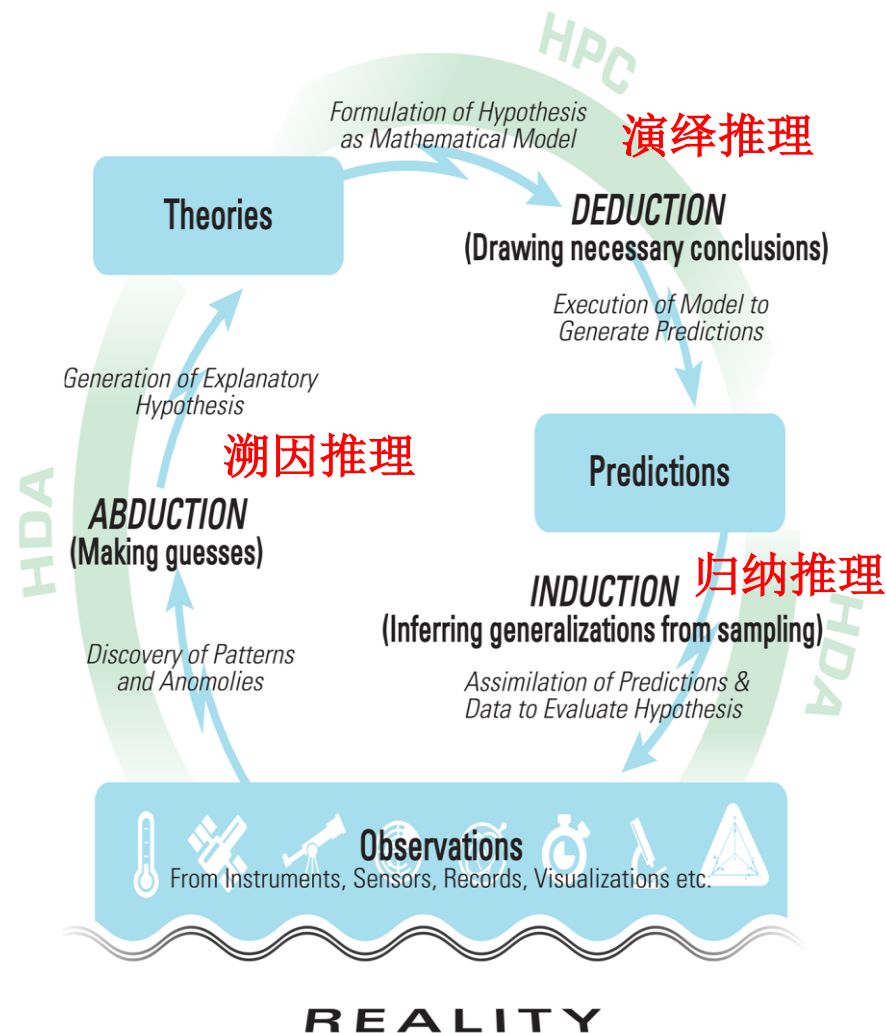
- 第五范式工作

- 观点与综述
- 科学智算实例（中科院计算所贾伟乐团队）

科学研究的范式演变（paradigm shift）

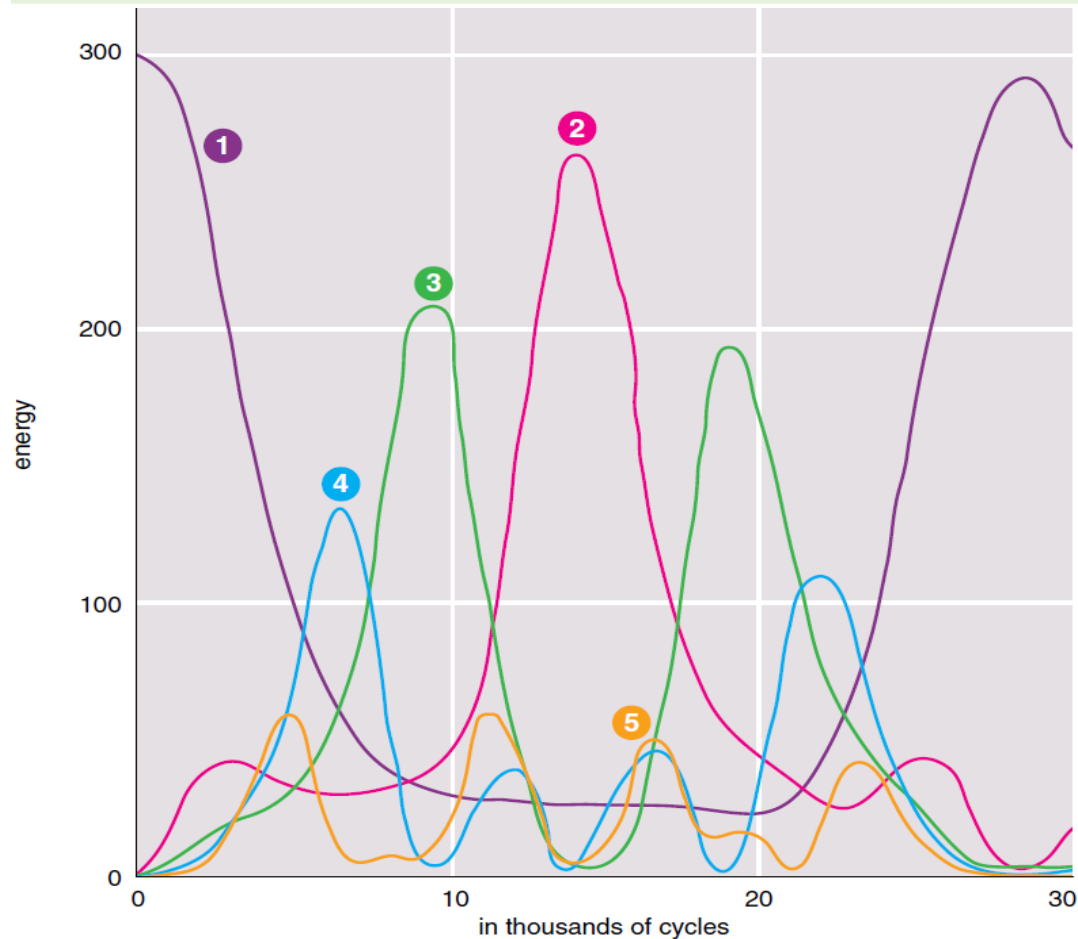
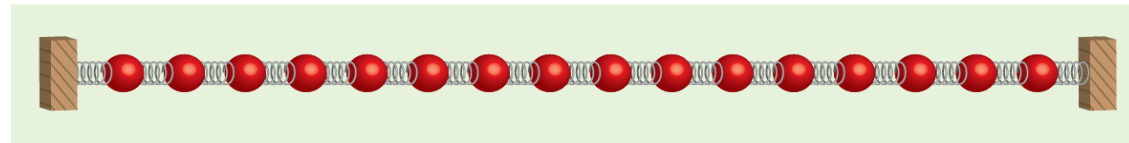
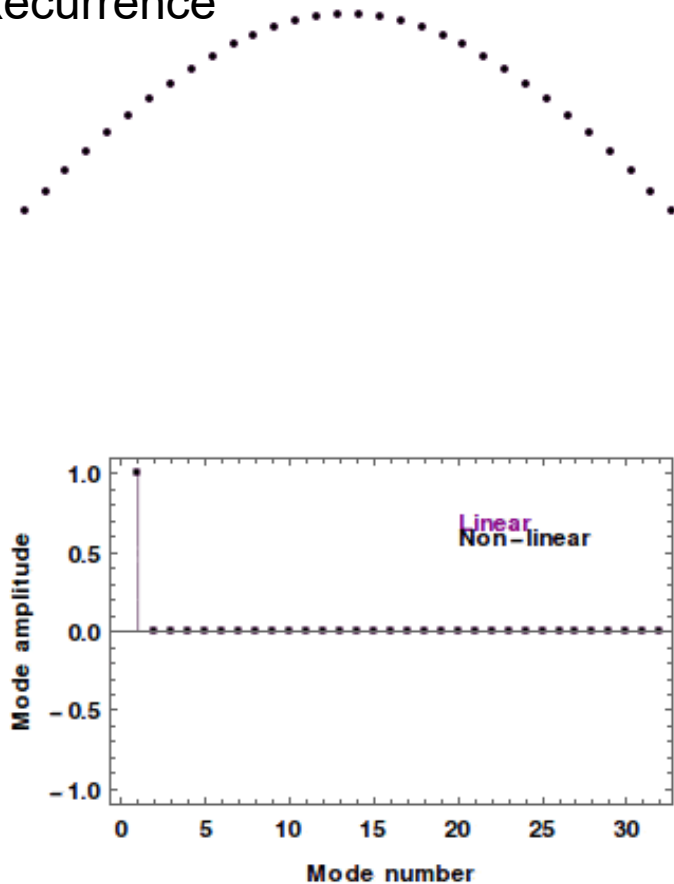
- 第一范式：理论研究
 - 牛顿、爱因斯坦
- 第二范式：实验观察
 - 伽利略
- 第三范式：计算机模拟
- 第四范式：数据处理
- 第五范式：AI辅助科学
 - AI4Science
 - 第一性原理模拟+深度学习+大数据
 - 智能超算：演绎推理+归纳推理+溯因推理

科学计算



3.1 费米等在1953年发明计算机模拟（第三范式）

- 也称为“数值实验”
 - 在Maniac计算机上求解
- Fermi–Pasta–Ulam悖论
 - 发现了FPU Recurrence



Porter, Mason A., et al. "Fermi, Pasta, Ulam and the Birth of Experimental Mathematics: A numerical experiment that Enrico Fermi, John Pasta, and Stanislaw Ulam reported 54 years ago continues to inspire discovery." *American Scientist* 97.3 (2009): 214-221.

3.2 第四范式：数据处理

● 黑洞照片实例

● 天文学依赖望远镜

- 亚米-米级
 - LAMOST: <8米
 - 郭守敬望远镜
- FAST: 500米
 - 中国天眼

● 黑洞拍照需要 直径1万千米的望远镜！



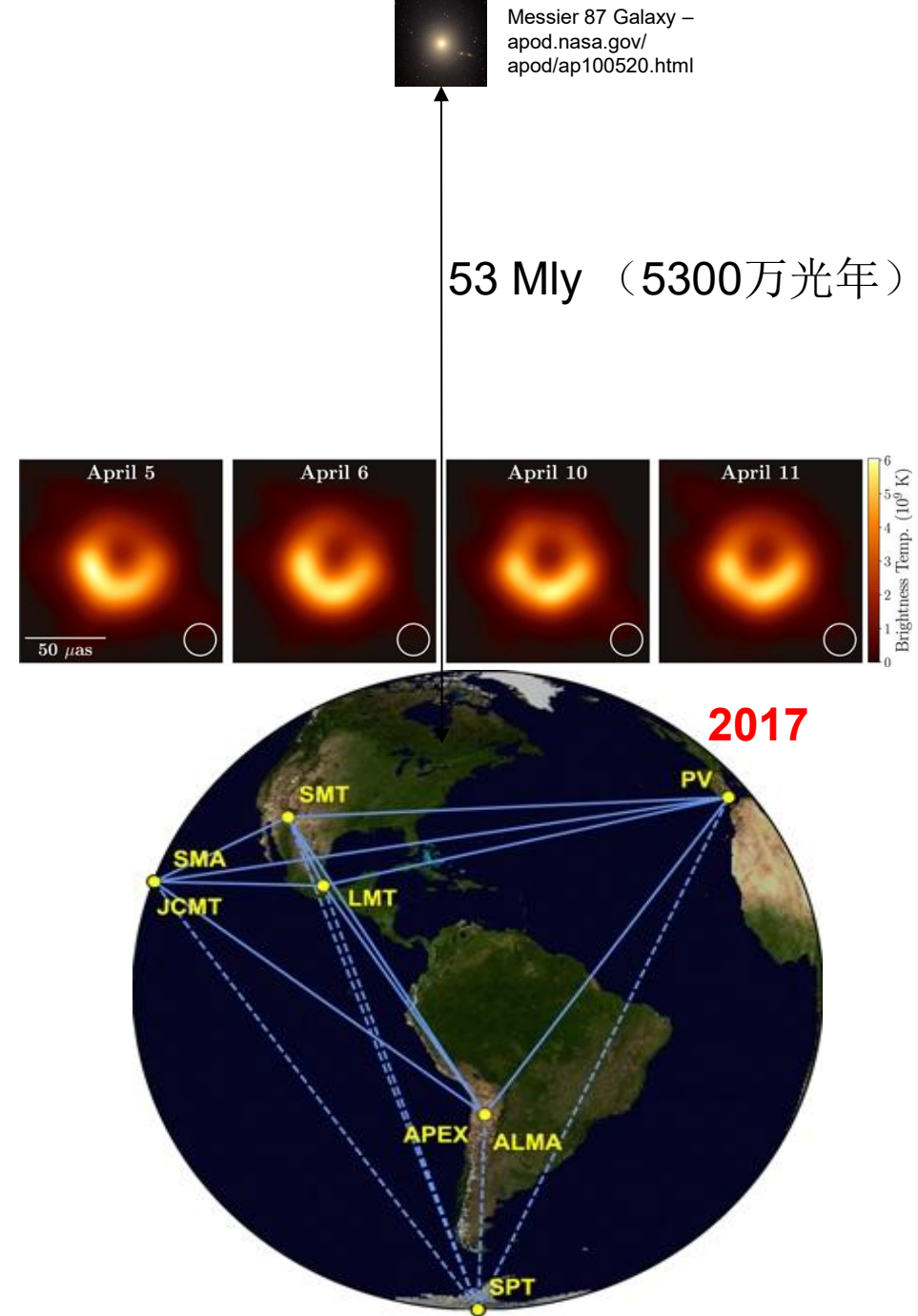
郭守敬望远镜



中国天眼

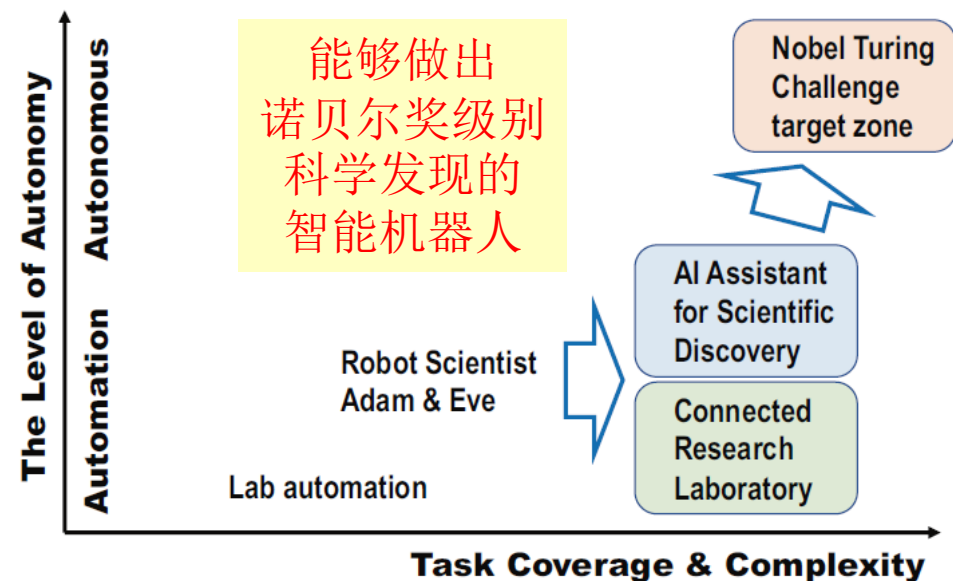
地球规模望远镜

- Event Horizon Telescope
 - 拍照（从曝光到生成照片）
 - 2017年4月，多次曝光
 - 采用原子钟同步
 - 将多点数据硬盘运到超级计算中心
 - 大约2年数据处理，生成黑洞照片
 - 2019年4月，发表论文
 - 验证了爱因斯坦广义相对论



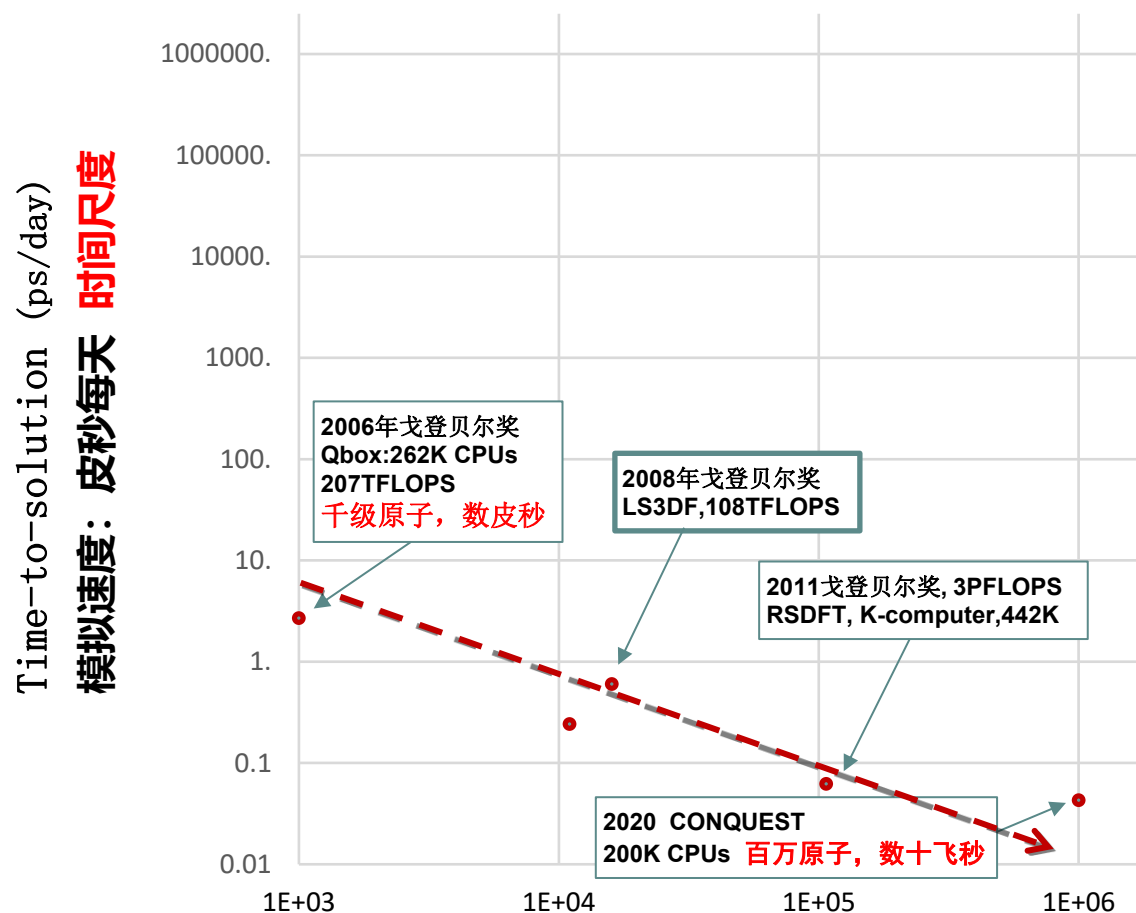
3.3 第五范式：AI辅助科学（AI4Science）

- 非常多的AI4S实际工作，汇聚成了几个主要学术观点
- AI4S带来多种好处
 - 提升计算速度，提高科研效率（人员时间与物质成本），提高科研质量，新的科教渠道
- AI4S正在改变机器的作用：从工具变为伙伴
 - **Nature Reviews Physics社评** 2022: Machines are no longer mere tools, but equal partners in scientific exploration, exchanging ideas, intuition and understanding with the human peers
- AI4S为科学带来新挑战
 - 不懂装懂，威胁多样性
- 五个范式常常混合使用
 - 计算：第一性原理模拟+数据处理+深度学习
- AI4S新倡议：诺贝尔图灵挑战



传统第一性原理分子动力学计算：十四年进展

空间尺度提高3个数量级，但时间尺度降低3个数量级

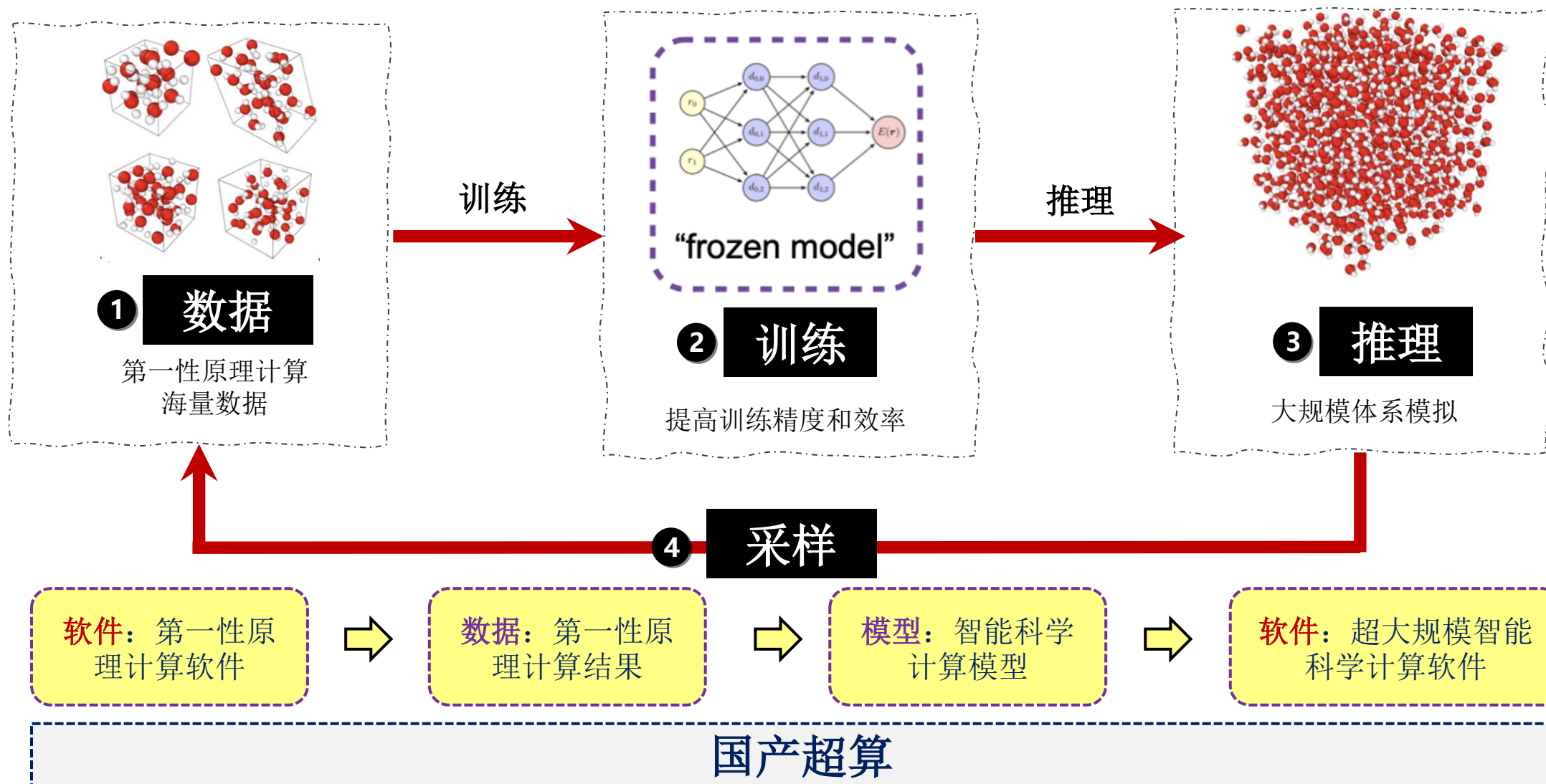


根本原因：计算量= $O(N^3)$
N是原子数目

系统规模N提升1000倍，计算量提升
 $1000 \times 1000 \times 1000 = 10$ 亿倍

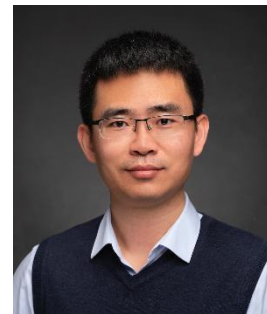
贾伟乐:

HPC+AI驱动的第一性原理分子动力学的四个步骤

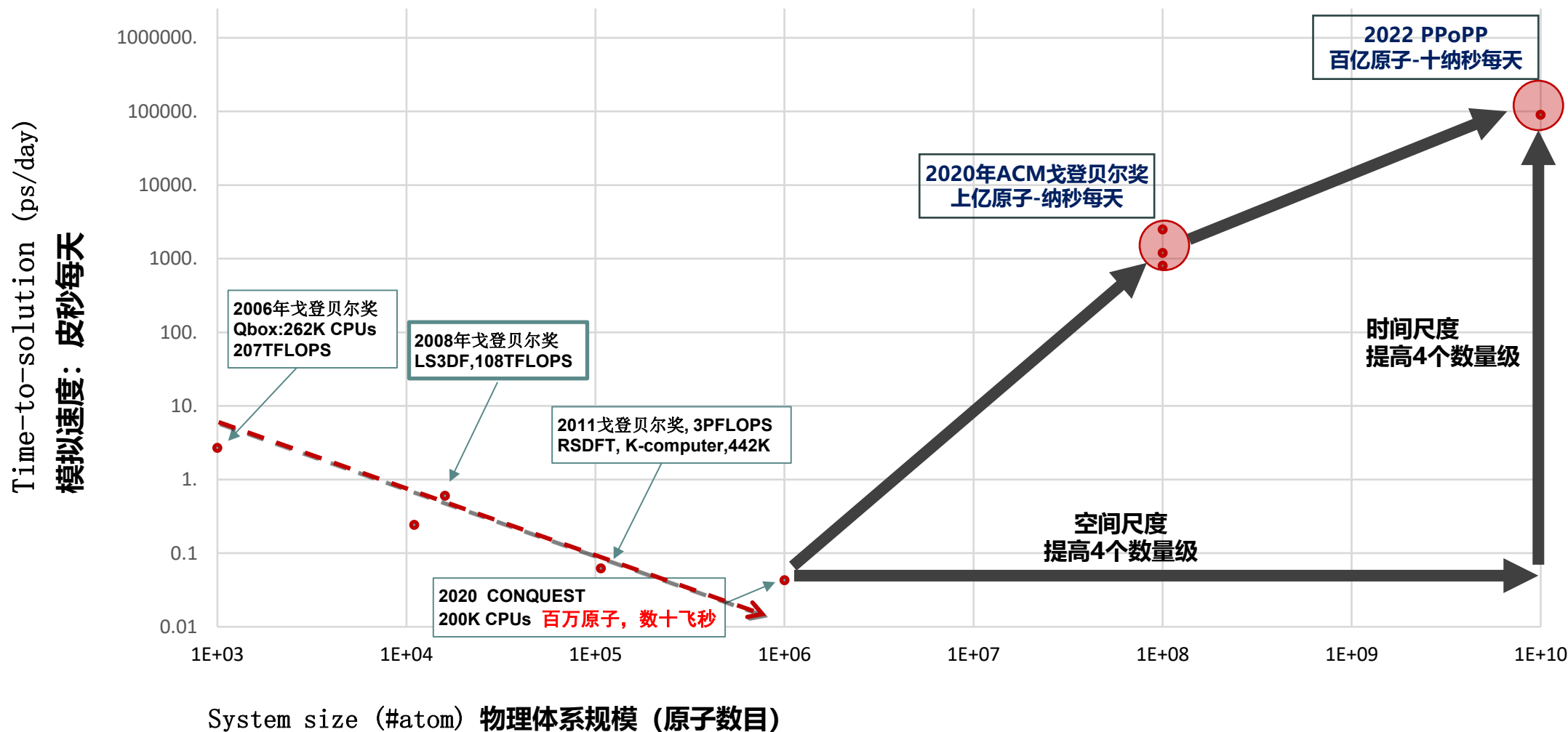


HPC+AI驱动的第一性原理分子动力学

效果：时空尺度提高4个数量级



贾伟乐研究员
中国科学院计算技术研究所
jiaweile@ict.ac.cn
<https://people.ucas.ac.cn/~jiaweile>



参考文献

● 综述与观点

- Hey, T., Tansley, S. & Tolle, K. The Fourth Paradigm: Data-Intensive Scientific Discovery (Microsoft Research, 2009).
- Editorial, Pervasive machine learning in physics, Nature Reviews Physics volume 4, p. 353 (2022).
- Wang, H. et al. Scientific discovery in the age of artificial intelligence, Nature, vol. 620, pp. 47–60 (2023).
- Douglas, MR. How will we do mathematics in 2030? CERN Colloquium, Sep 30, 2021 <https://indico.cern.ch/event/1077061/attachments/2320414/3951088/decade19.pdf>
- 贾伟乐. HPC+AI在第一性原理分子动力学中的应用, 科学计算技术与应用会议, 2023, 中国-三亚
- 诺贝尔图灵挑战倡议 <https://www.nobelturingchallenge.org/>

● AI4Science的新挑战与社会进步

- Editorial, Defining physicists' relationship with AI, Nature Reviews Physics volume 4, page 735 (2022)
- Lisa Messeri & M. J. Crockett, Artificial intelligence and illusions of understanding in scientific research, Nature, volume 627, pages 49–58 (2024)

4. 21世纪文明需要创新型人才

应对“百年未有之大变局”，需要世纪眼光

- 21世纪文明是“发展文明”
- 21世纪文明是“信息文明”

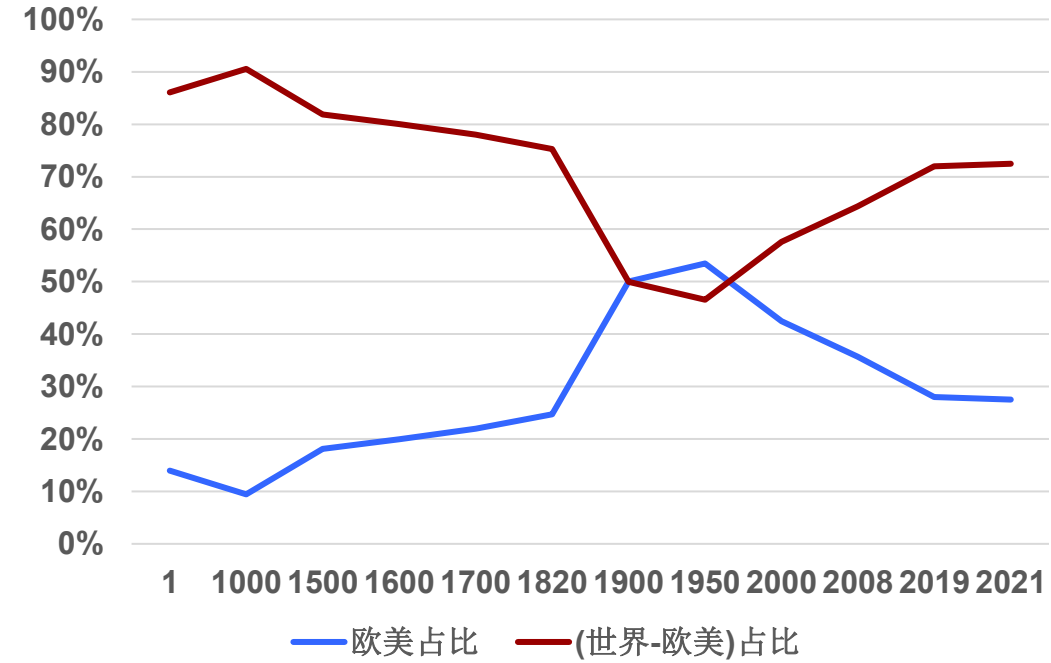
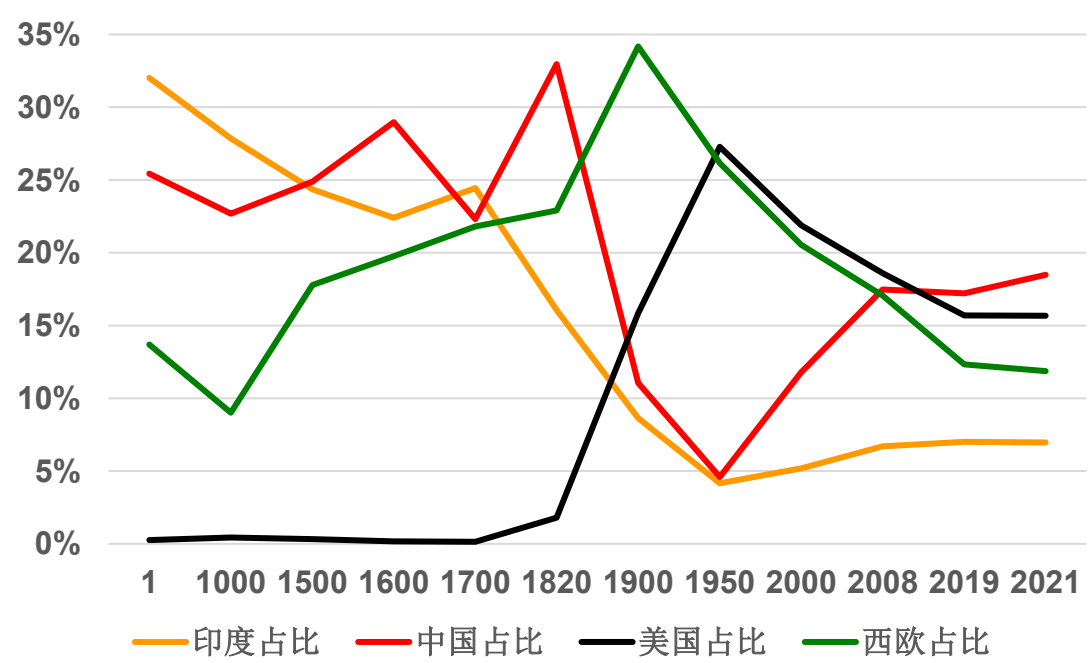
放眼世界，我们面对的是百年未有之大变局。新世纪以来一大批新兴市场国家和发展中国家快速发展，世界多极化加速发展，国际格局日趋均衡，国际潮流大势不可逆转。

习近平，2017

21世纪文明是“国家发展文明”

- 放眼世界， 20世纪文明的最大特征是什么？
- 20世纪文明的最大特征是殖民主义的消亡
 - 是“民族解放”、“民族独立”、“主权自由”

彼得卡赞斯坦的观点



数据来源：世界银行 <https://datatopics.worldbank.org/world-development-indicators/>; Angus Maddison, The World Economy, OECD 2006; <https://www.rug.nl/ggdc/historicaldevelopment/maddison/releases/maddison-project-database-2023>

如果认同人类文明发展的长期趋势是“人类不断解放”

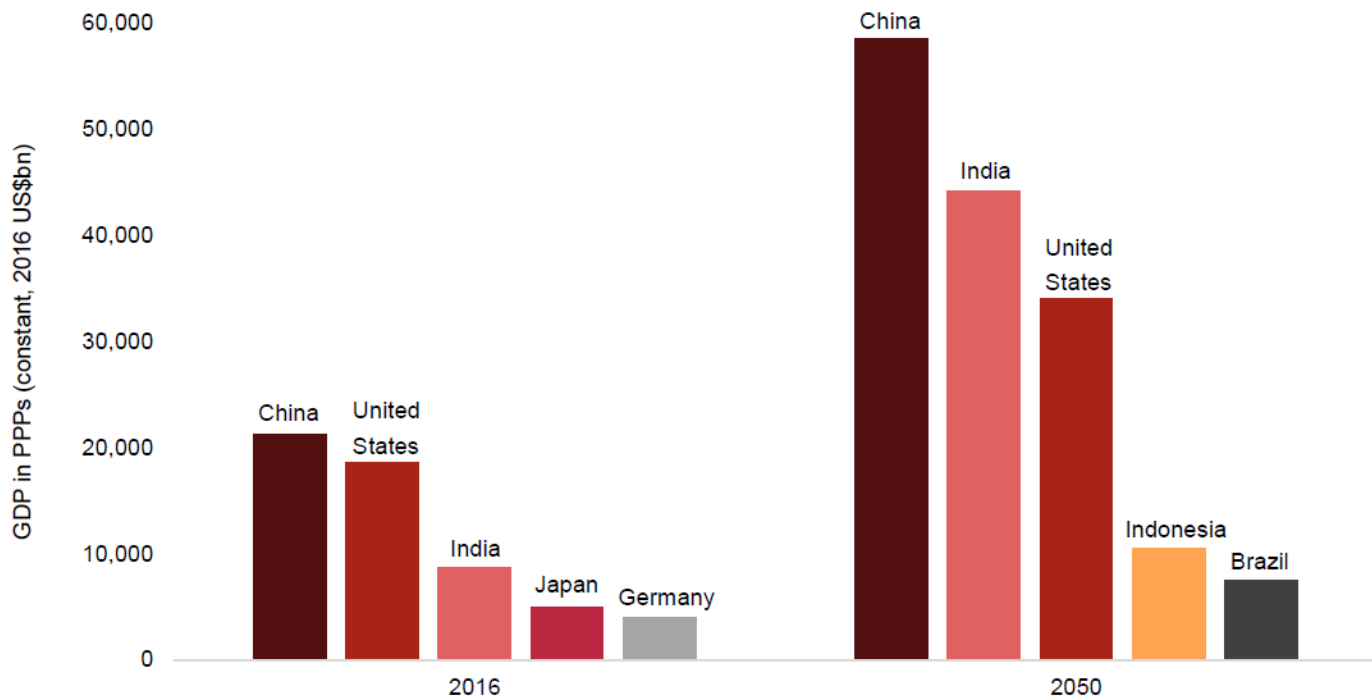
- 自由解放的表征：摆脱 ①野蛮；②神权、皇权、列强、奴役制度；③垄断控制
- 20世纪文明的最大特征是殖民主义的消亡
 - 发展中国家享有了“民族独立”； 20世纪文明是追求“主权自由”的文明
- 21世纪文明的最大特征是所有国家发展，尤其是发展中国家科技兴起
 - “主权自由” → “国家发展的自由”

年度	GDP（万亿国际元）			人均GDP（国际元）			人口平均预期寿命（岁）		
	世界	美国	中国	世界	美国	中国	世界	美国	中国
1820	0.69	0.01	0.23	666	1,257	600	26	39	24
1900	1.97	0.31	0.22	1,261	4,091	545	31	47	24
1950	5.34	1.46	0.24	2,111	9,561	448	49	68	41
2000	36.69	8.03	4.32	6,038	28,467	3,421	68	77	72
2008	50.97	9.49	8.91	7,614	31,178	6,725	70	78	75
2019	130.66	20.51	22.49	16,877	62,471	15,978	73	79	78
2021	134.82	21.13	24.94	17,091	63,636	17,658	71	76	78

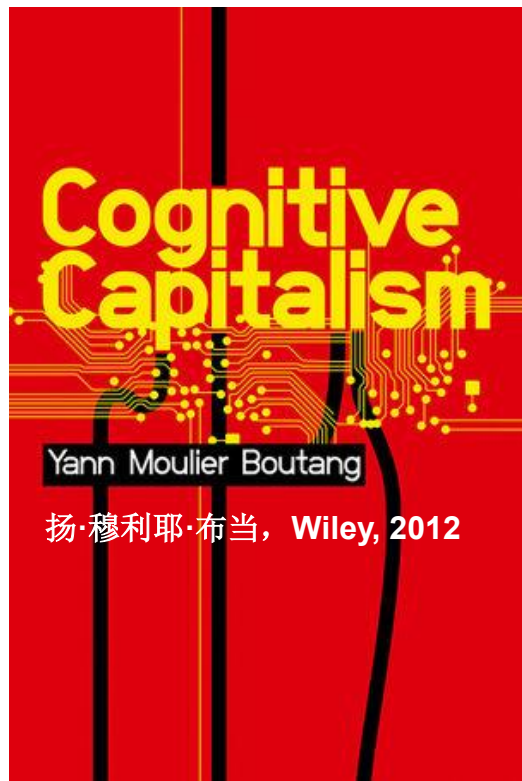
数据来源：世界银行 <https://datatopics.worldbank.org/world-development-indicators/>；Angus Maddison, The World Economy, OECD 2006；
<https://www.rug.nl/ggdc/historicaldevelopment/maddison/releases/maddison-project-database-2023>

如果认同人类文明发展的长期趋势是“人类解放事业”

- 21世纪文明的最大特征是所有国家发展，尤其是发展中国家兴起
 - “主权自由” → “发展的自由”；必然要追求“科技发展的自由”
 - 市场的四个阶段：贸易资本主义 → 工业资本主义 → 金融资本主义 → 认知资本主义



数据来源：普华永道，The World in 2050报告，2017



中国进入新发展阶段

尤其需要懂计算思维的创新人才

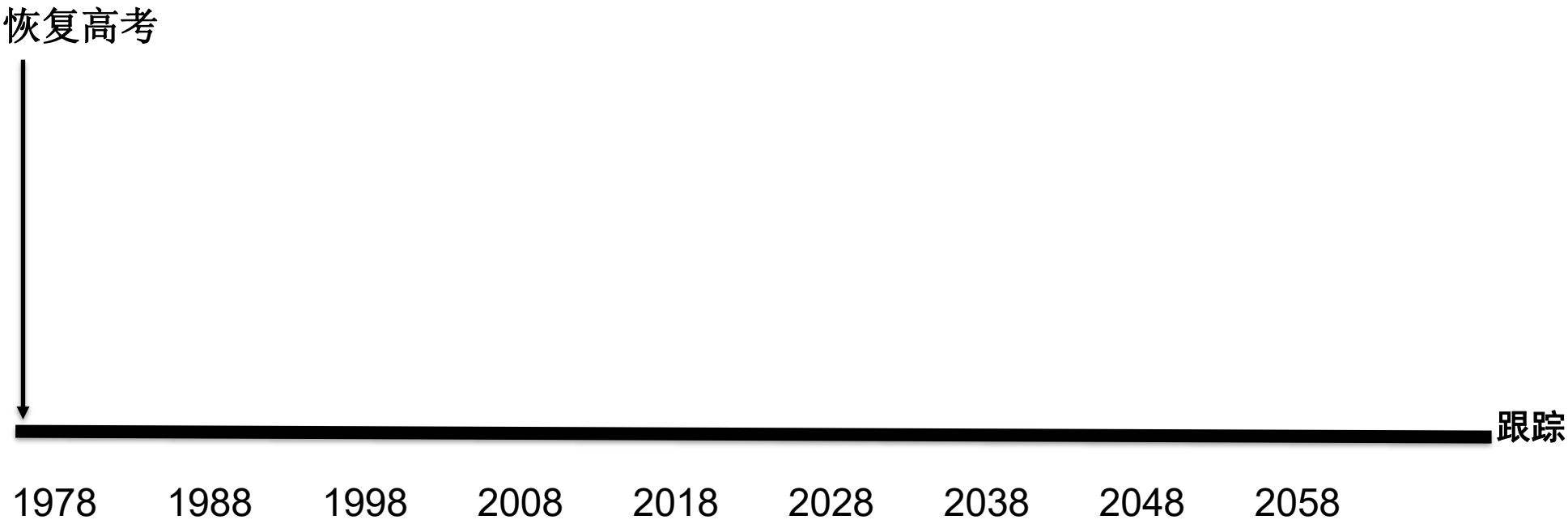
“0到1创新”是新质生产力的核心

什么是新发展阶段？

跟踪-追赶-贡献的贡献阶段

跟踪阶段：毕业生的主流出口是出国、外企

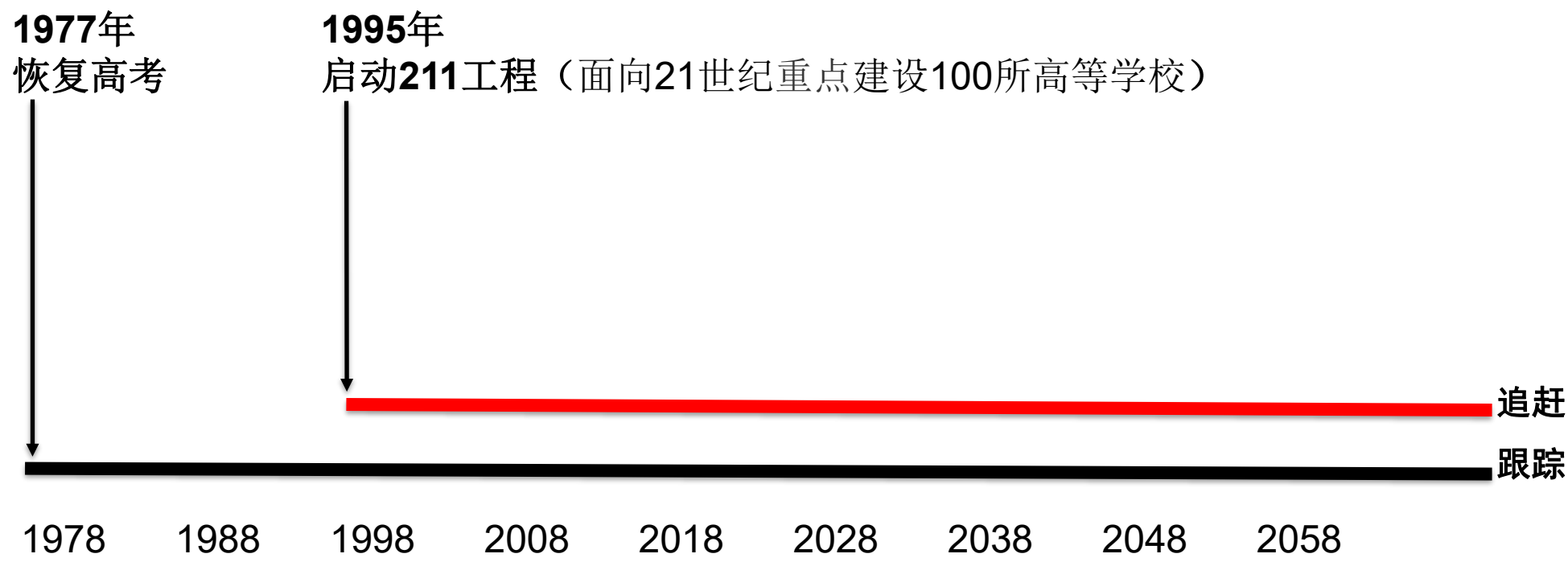
Follow



什么是新发展阶段？

跟踪阶段：毕业生出口——追求是出国、外企
追赶阶段：

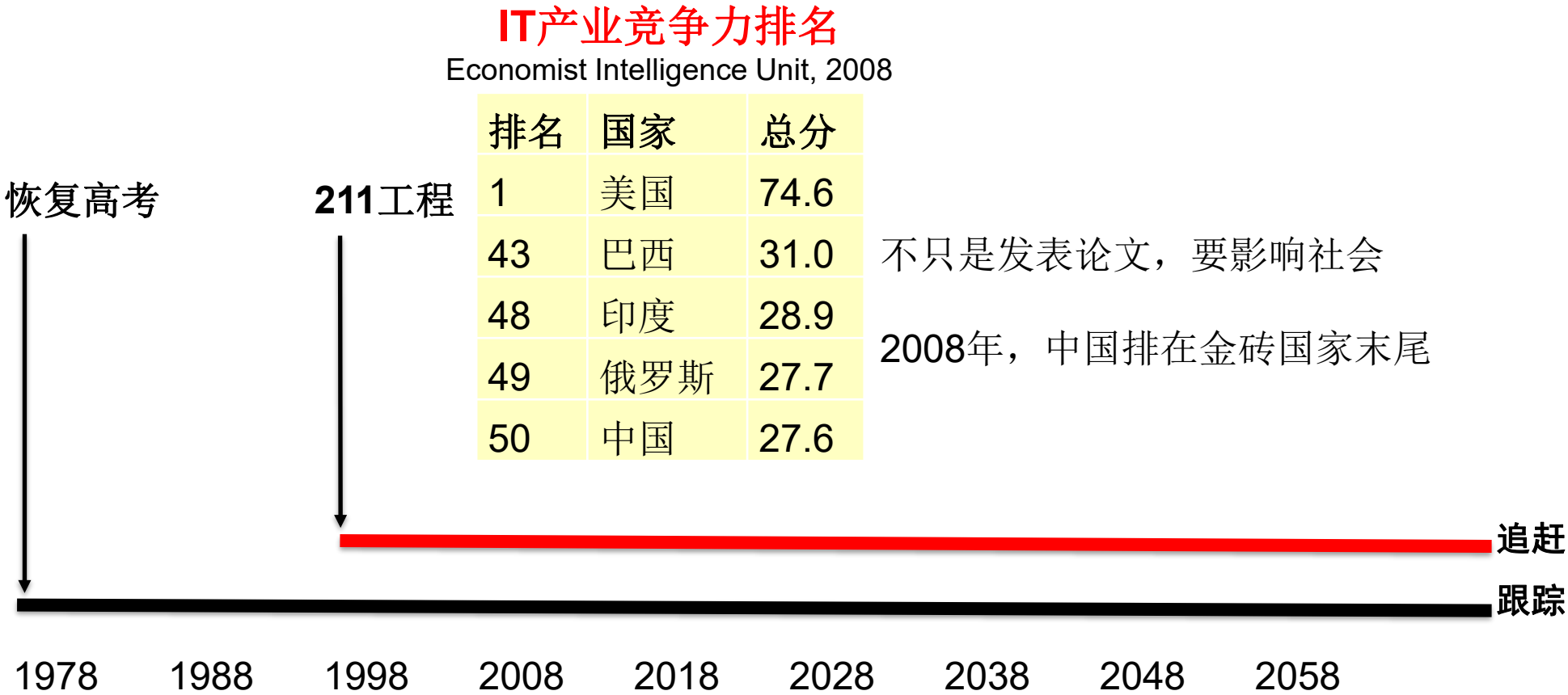
Follow
Catch up



什么是新发展阶段？

跟踪阶段：毕业生出口——追求出国、外企
追赶阶段：毕业生出口——逐渐过渡到大厂、研究所名校

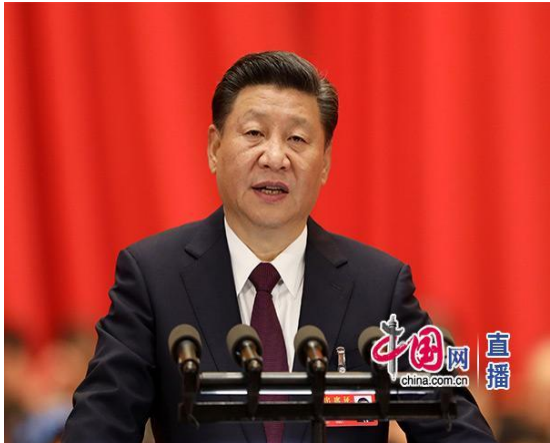
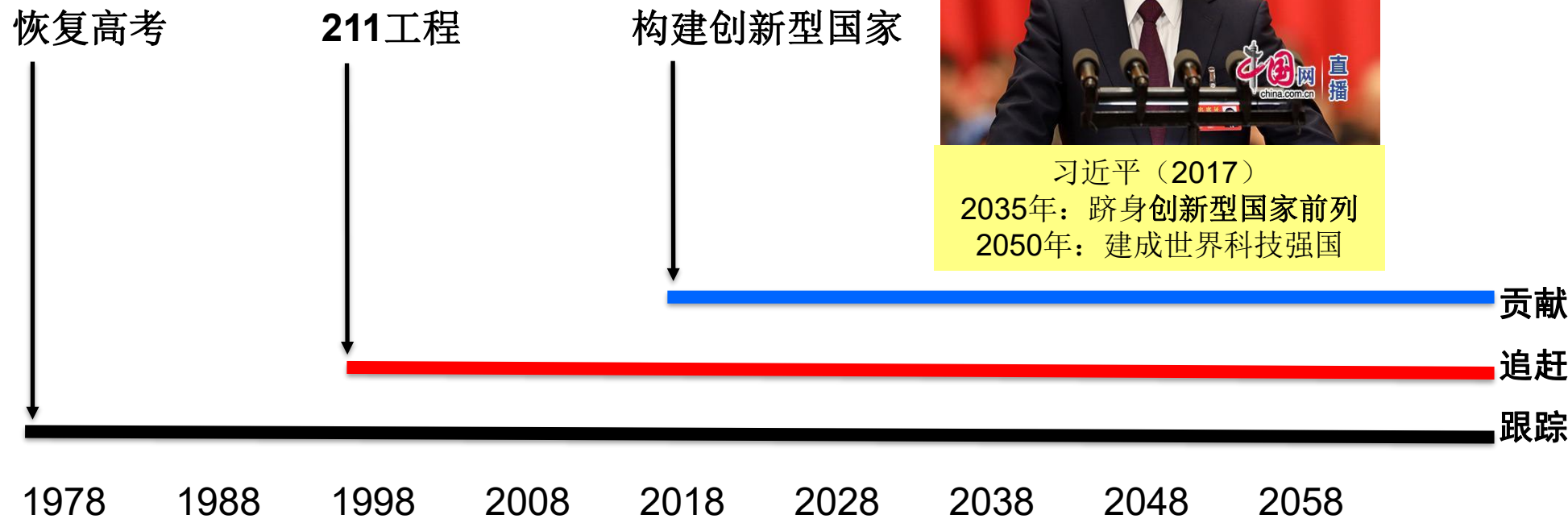
Follow
Catch up



新发展阶段是为全球做贡献的阶段

- 跟踪阶段：毕业生的主流出口是出国和外企 Follow
- 追赶阶段：毕业生的主流出口是大厂、名校 Catch up
- 贡献阶段：毕业生的出口是创新型多样化 Contribute, Lead

贡献阶段的大学教育，与跟踪追赶的大学教育，是不一样的

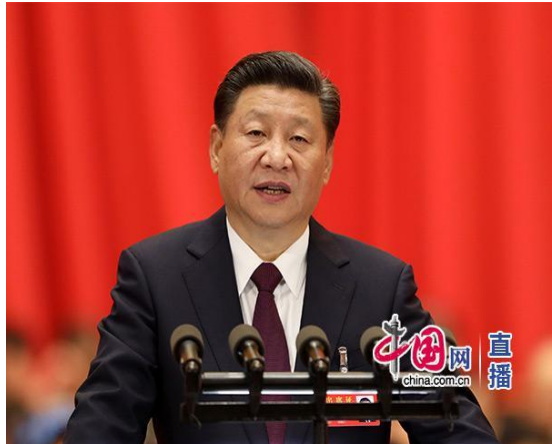
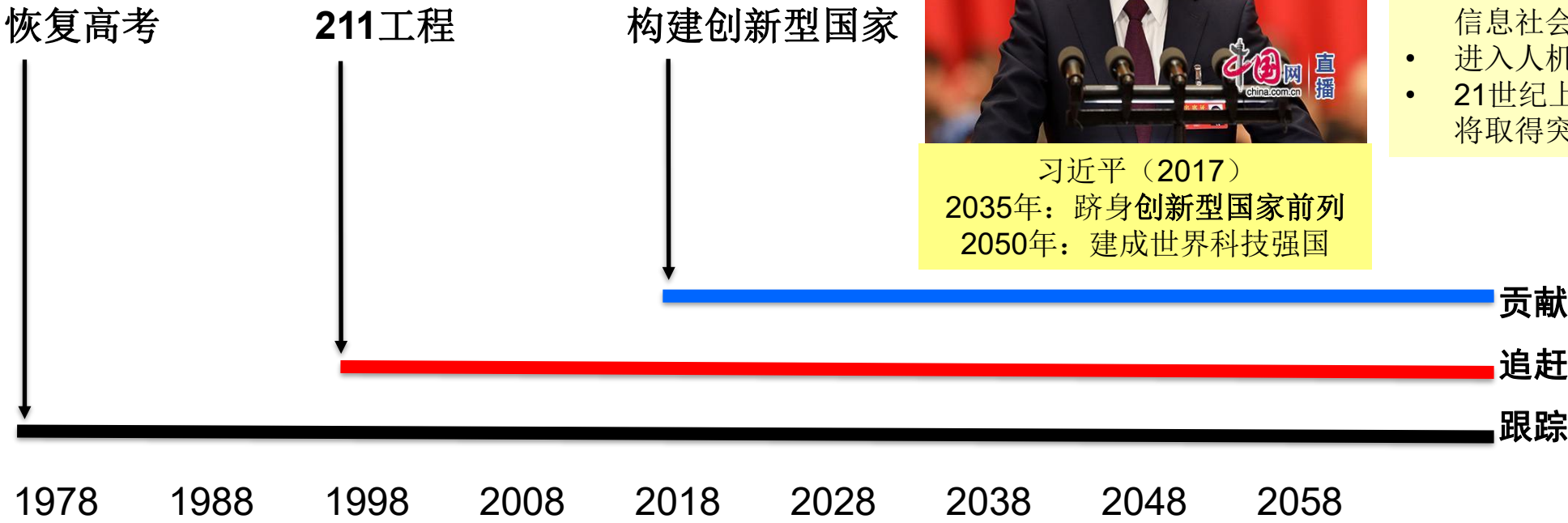


习近平（2017）
2035年：跻身创新型国家前列
2050年：建成世界科技强国

新发展阶段是为全球做贡献的阶段

- 跟踪阶段：毕业生的主流出口是出国和外企
- 追赶阶段：毕业生的主流出口是大厂、名校
- 贡献阶段：毕业生的出口是创新型多样化
- Follow
- Catch up
- Contribute, Lead

贡献阶段的大学教育，与跟踪追赶的大学教育，是不一样的
信息社会的大学教育，与工业社会的大学教育，是不一样的



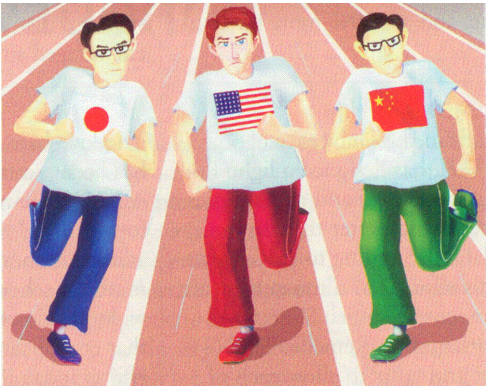
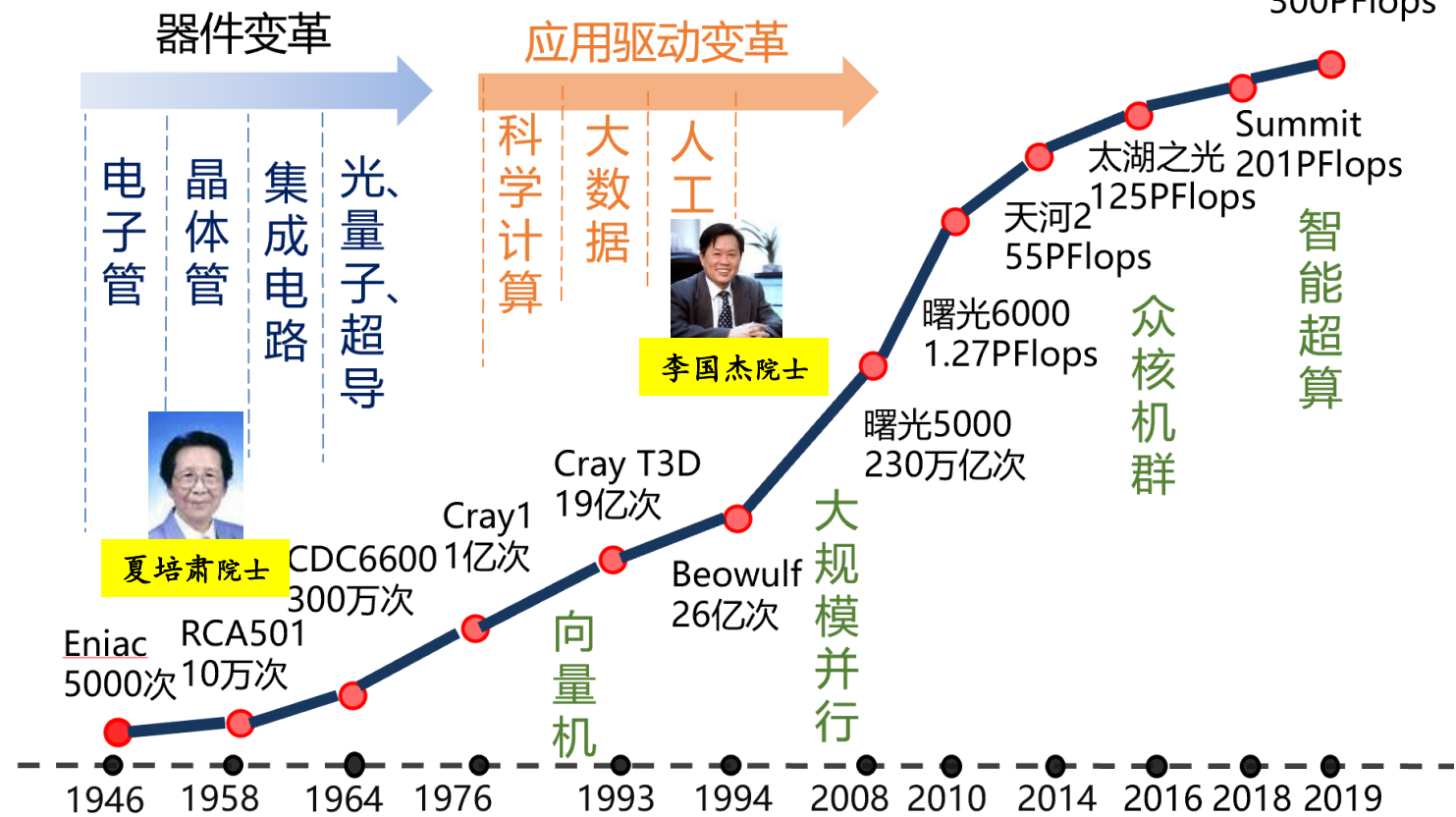
习近平（2017）
2035年：跻身创新型国家前列
2050年：建成世界科技强国



- 世界将在2050年进入信息社会中高级阶段
- 进入人机物三元世界
- 21世纪上半叶信息科学将取得突破性进展

继承敢为人先的勇气

超级计算机：从1946到2023



CACM, 2005.11
“Super Competition”

- Top500是美国和德国科学家设立的榜单，每年两次发布，用高斯消元法解方程的**计算速度（算力）**排名，列出全球最快的500台超级计算机
- 中国超级计算机的计算速度，从Top500不入榜，进步到领先行列
- 2014年，中科曙光A股上市
- 地缘政治竞争
- 2022年，曙光系上市公司市值超5000亿元



1951年回国到清华电子系
1953年到科学院做计算机

1978年到中科院计算所，
1987年回国

1989年北大毕业到
中科院计算所



中国科学院大学
University of Chinese Academy of Sciences

计算机科学导论

期末复习

徐志伟 zxu@ict.ac.cn

提纲

- 计算机学科的研究对象
- 计算机学科的基本研究方法
 - 计算思维基本解题思路：PEPS（活力方法论）
 - 对计算思维的十个理解：**Acu-Exams-CP**
- 逻辑思维
 - 建模既需要心胸开阔，又需要科学严谨
 - 图灵机的通用性和局限性
- 算法思维
 - 什么是“构造性”：**Acu-Exams-CP**之**E**（Effectiveness）
 - 掌握高德纳算法定义，理解渐近记号
- 系统思维的“以一耦万”抽象栈思想
 - 信息隐藏程序回顾：自顶向下设计
 - 用时序电路设计4位串行加法器、减法器
- 网络拓扑：第二代搜索引擎示例
- 协议栈：互联网通信示例

1. 计算机科学的研究对象

比照《计算机科学基础报告》

- “计算机科学是研究计算机以及它们能干什么的一门学科。它研究**抽象计算机**的能力与局限，**真实计算机**的构造与特征，以及用于求解问题的无数**计算机应用**。”
 - 计算机科学涉及**符号**及其操作；
 - 关注多种**抽象**概念的创造和操作；
 - 创造并研究**算法**；
 - 创造各种**人工结构**，尤其是不受物理定律限制的结构；
 - 利用并应对**指数增长**；
 - 探索计算能力的**基本极限**；
 - 关注与人类**智能**相关的复杂的、分析的、理性的活动。

知道概念，能够举例说明其特点

笔记本电脑

有限状态机、图灵机、
冯诺依曼模型

- “计算机科学是研究计算机以及它们能干什么的一门学科。它研究**抽象计算机**的能力与局限，**真实计算机**的构造与特征，以及用于求解问题的无数**计算机应用**。”
- 计算机科学涉及**符号**及其操作；
- 关注多种**抽象**概念的创造和操作；
- 创造并研究**算法**；
- 构造各种**人工结构**；
- 用并应对**指数增长**；
- 探索计算能力的**基本极限**；
- 关注与人类**智能**相关的复杂的、分析的、理性的活动。

快速排序程序

各种编程抽象和系统抽象

逻辑推理
命题逻辑
谓词逻辑

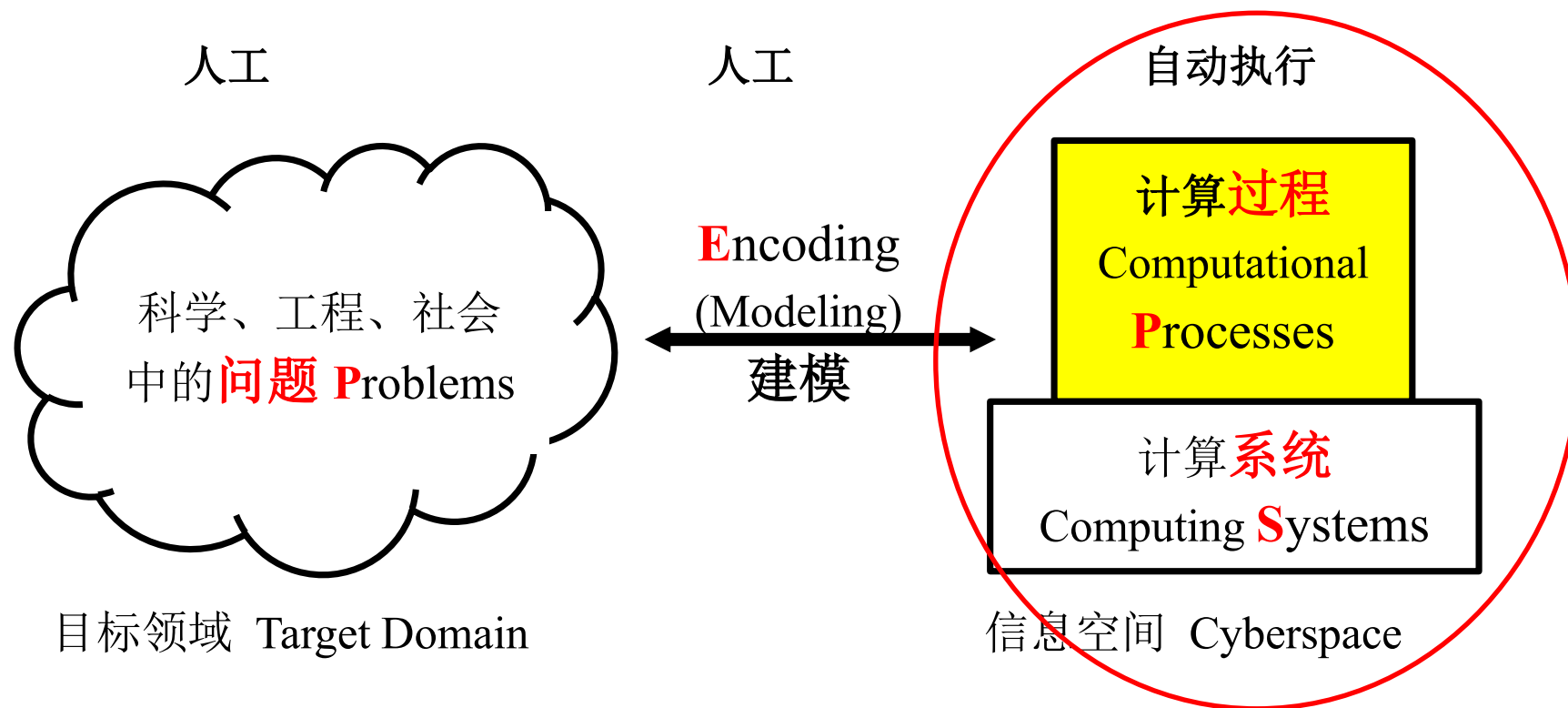
个人作品

斐波那契递归程序
P与NP

图灵机不可计算问题
哥德尔不完备定理

2. 计算思维基本方法：活力解题法（PEPS）

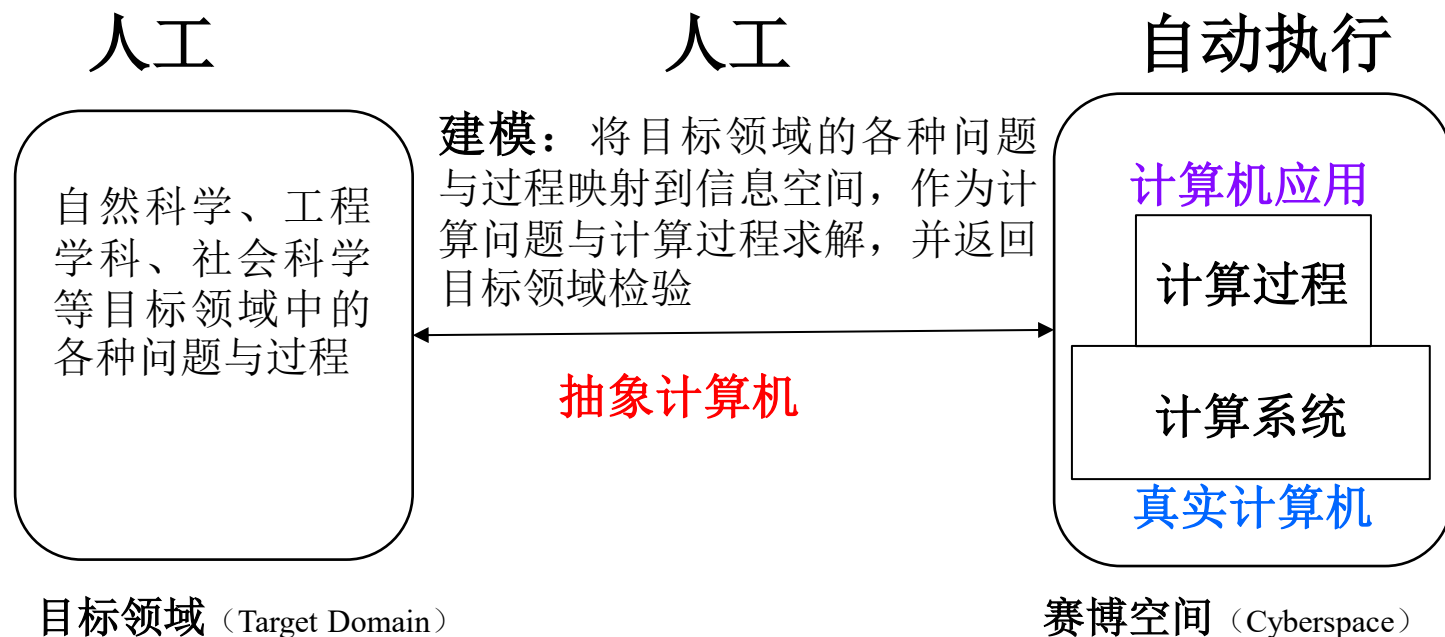
- 定义领域**问题**（**P**roblem in target domain）
- **建模**（**E**ncoding）到信息空间（赛博空间，cyberspace）的计算问题
- 自动执行计算**系统**（**S**ystem）上的计算**过程**（**P**rocess），解决问题
- 映射回到目标领域检验



计算思维基本方法

- 一种认识世界、定义问题、解决问题的科学方法
 - 计算系统 = 计算机
 - 使用算法、程序、状态转移表等刻画计算过程
- 建模是双向映射
- 目标领域也可以是赛博空间

计算机科学研究**抽象计算机**，**真实计算机**，**计算机应用**。



计算思维概念

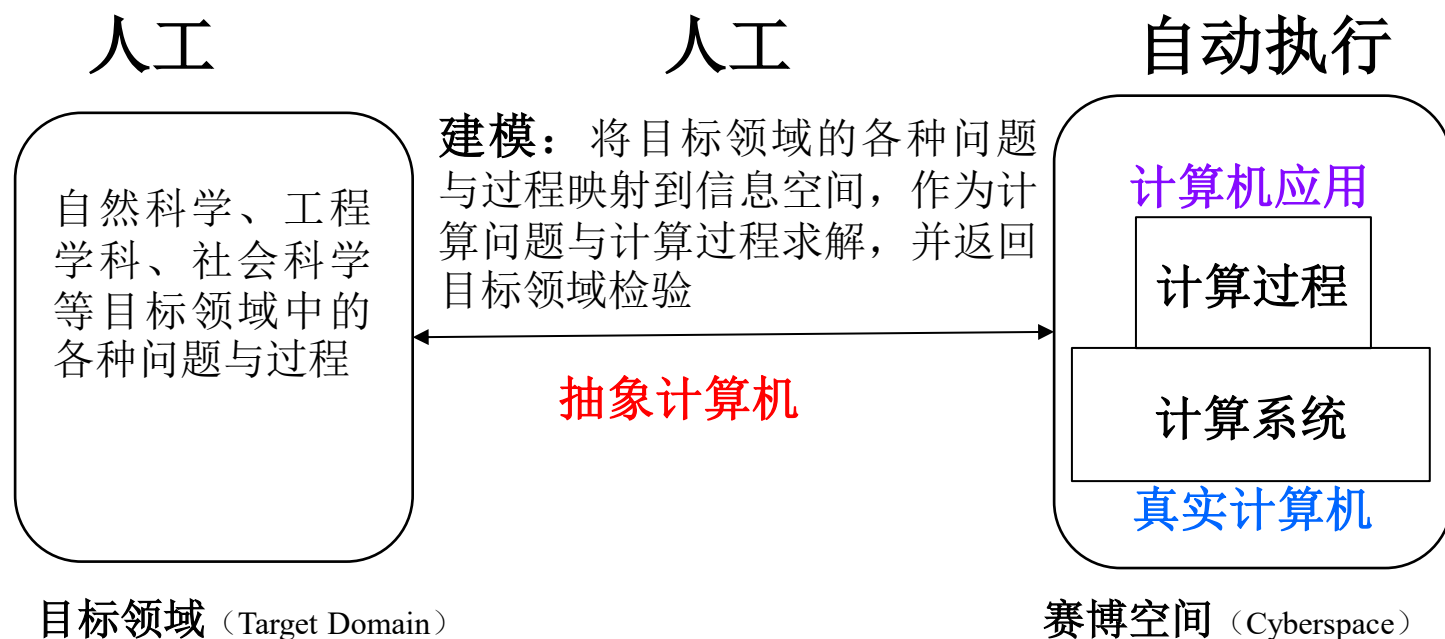
对计算思维的十个理解: **Acu-Exams-CP**

计算机科学是研究计算过程的科学

(A): Automatically execution	自动执行。计算机能够自动执行由离散步骤组成的计算过程。
(C): Correctness	正确性。计算机求解问题的正确性往往可以精确地定义并分析。
(U): Universality	通用性。计算机能够求解任意可计算问题。
(E): Effectiveness	构造性。人们能够构造出聪明的方法让计算机有效地解决问题。
(X): Complexity	复杂度。这些聪明的方法（算法）具备时间/空间复杂度。
(A): Abstraction	抽象化。少数精心构造的计算抽象可产生万千应用系统。
(M): Modularity	模块化。多个模块有规律地组合成为计算系统。
(S): Seamless Transition	无缝衔接。计算过程在计算系统中流畅地执行。
(C): Connectivity	连接性。很多问题涉及用户/数据/算法的连接体，而非单体。
(P): Protocol Stack	协议栈。连接体的节点之间通过协议栈交互。

3. 逻辑思维

- 逻辑思维关注计算过程的正确性和计算机的通用性
- 不止是只关注已经描述成算法或程序的计算过程PEPS，建模阶段（PEPS）也极其重要
 - 既要心胸开阔，也要科学严谨



3.1 存在很多科学建模方法

科学需要心胸开阔，切忌鼠目寸光、把现有的科学知识当成教条
存在很多逻辑论证（arguments）方法

下述陈述句序列是不是一个逻辑论证 **logical argument**?

- 屠呦呦1930年出生于宁波。
- 她的长辈从《诗经》的“呦呦鹿鸣食野之蒿”给她取名“呦呦”。
- “呦呦鹿鸣食野之蒿”之中的“蒿”就是青蒿。
- 屠呦呦长大后研究出了青蒿素，一种治疗疟疾的药物。
- 《诗经》预言成真，冥冥中自有定数。

下述陈述句是不是一个逻辑论证 **logical argument**?

- 屠呦呦1930年出生于宁波。
- 她的长辈从《诗经》的“呦呦鹿鸣食野之蒿”给她取名“呦呦”。
- “呦呦鹿鸣食野之蒿”之中的“蒿”就是青蒿。
- 屠呦呦长大后研究出了青蒿素，一种治疗疟疾的药物。
- 网上有人言：《诗经》预言成真，冥冥中自有定数。这纯属胡说八道。

下述陈述句是不是一个逻辑论证 logical argument?

- 屠呦呦研究青蒿素，参考了葛洪的《肘后备急方》一书内容。
- 葛洪是炼丹的道士，信奉神仙，其代表作《抱朴子》是讲修仙的书。
- 屠呦呦研究青蒿素的工作，引用了讲迷信的“葛神仙”的著作。
- 屠呦呦研究青蒿素的工作，不科学。



《肘后备急方》

卷三

治寒热诸疟方第十六

治疟病方。鼠妇，豆豉二七枚，合捣令相和。未发时服二丸，欲发时服一丸。

又方：**青蒿一握。以水二升渍，绞取汁。尽服之。**

又方：用独父蒜于白炭上烧之，末。服方寸匕。

又方：五月五日，蒜一片（去皮，中破之，刀割），合容巴豆一枚（去心皮，纳蒜中，令合）。以竹挟以火炙之，取可热，捣为三丸。未发前服一丸。不止，复与一丸。

又方：取蜘蛛一枚，芦管中密塞，管中以绀颈，过发时乃解去也。

又方：**日始出时，东向日再拜，毕正长跪，向日叉手，当闭气，以书墨注其管两耳中，各七注，又丹书舌上，言子曰死，毕，复再拜，还去勿顾，安卧勿食，过发时断，即瘥。**

.....



下述陈述句是不是一个逻辑论证 logical argument?

- 屠呦呦研究出了青蒿素，一种治疗疟疾的药物。
 - 屠呦呦研究青蒿素，参考了葛洪的《肘后备急方》一书内容。
 - 葛洪是炼丹的道士，信奉神仙，其代表作《抱朴子》是讲修仙的书。
 - 屠呦呦研究青蒿素的工作，引用了讲迷信的“葛神仙”的著作。
 - 屠呦呦研究青蒿素的工作，不科学。
 - 疟疾患者不应该使用青蒿素。
-
- The Nobel Prize in Physiology or Medicine 2015 was awarded to William C. Campbell, Satoshi Ōmura and Youyou Tu.
 - 2015年度**诺贝尔生理学或医学奖**被授予三位研究者。奖金的一半授予**威廉·坎贝尔**和**大村智**，以表彰他们在创新蛔虫疗法方面的贡献；另一半授予**屠呦呦**，以表彰她在治疗疟疾方面的贡献。……屠呦呦制成了青蒿素，这种药物显著降低了疟疾患者的死亡率。

存在很多逻辑论证方法，科学中常常融合三种推理（Reasoning）

- Deductive reasoning 演绎推理
 - 从前提（premises，也称为论据）依据推理规则（inference rules）推导出结论（conclusion）
 - 好的演绎推理逻辑论证称为有效的（valid）
 - 不能保证结果是正确的
 - 如果前提都是正确的，结论肯定也是正确的
 - 如果前提有错误，结论可能也是错误的
- Inductive reasoning 归纳推理
 - 从现有样本的结果，推导出泛化结论
- Abductive reasoning 溯因推理
 - 根据理论和观察猜测出最好的假说
- 本课程聚焦**可自动执行的**演绎推理



3.2 建模需要严谨性

- 给定下列两个论据，能推出结论是成立的吗？
 - 论据1：所有上过计科导课的同学都精通Go语言。
 - 论据2：某些精通Go语言的同学可成为下一年计科导课助教。
 - 结论：有些上过计科导的同学可成为下一年计科导课助教。

建模需要严谨性

- 给定下列两个论据，能推出结论是成立的吗？
 - 论据1：所有上过计科导课的同学都精通Go语言。
 - 论据2：某些精通Go语言的同学可成为下一年计科导课助教。
 - 结论：有些上过计科导的同学可成为下一年计科导课助教。
- 定义下列断言记号
 - $CS101(x)$ ：同学 x 上过计科导
 - $MasterGo(x)$ ：同学 x 精通Go语言
 - $TA(x)$ ：同学 x 可成为下一年计科导课助教

建模需要严谨性

- 给定下列两个论据，能推出结论是成立的吗？
 - 论据1：所有上过计科导课的同学都精通Go语言。
 - 论据2：某些精通Go语言的同学可成为下一年计科导课助教。
 - 结论：有些上过计科导的同学可成为下一年计科导课助教。
- 定义下列断言记号
 - CS101(*x*)：同学*x*上过计科导
 - MasterGo(*x*)：同学*x*精通Go语言
 - TA(*x*)：同学*x*可成为下一年计科导课助教

错误的建模

$\forall x[\text{MasterGo}(x)]$
 $\exists x[\text{MasterGo}(x) \rightarrow \text{TA}(x)]$
 $\exists x[\text{CS101}(x) \rightarrow \text{TA}(x)]$

论据2的谓词表达式应该是 $\exists x[\text{MasterGo}(x) \wedge \text{TA}(x)]$

MasterGo(<i>x</i>)	TA(<i>x</i>)	MasterGo(<i>x</i>) → TA(<i>x</i>)	MasterGo(<i>x</i>) ∧ TA(<i>x</i>)
0	0	1	0
0	1	1	0
1	0	0	0
1	1	1	1

建模需要严谨性

- 给定下列两个论据，能推出结论是成立的吗？
 - 论据1：所有上过计科导课的同学都精通Go语言。
 - 论据2：某些精通Go语言的同学可成为下一年计科导课助教。
 - 结论：有些上过计科导的同学可成为下一年计科导课助教。
- 严谨地构建谓词表达式、使用推理规则
 - 谓词逻辑往往不关心原初论据本身的正确性。即，假设给定论据是正确的。
 - 定义下列断言记号
 - $CS101(x)$ ：同学 x 上过计科导
 - $MasterGo(x)$ ：同学 x 精通Go语言
 - $TA(x)$ ：同学 x 可成为下一年计科导课助教

错误的建模

$\forall x[MasterGo(x)]$

$\exists x[MasterGo(x) \rightarrow TA(x)]$

$\exists x[CS101(x) \rightarrow TA(x)]$

论据1： $\forall x[CS101(x) \rightarrow MasterGo(x)]$

论据2： $\exists x[MasterGo(x) \wedge TA(x)]$

结论： $\exists x[CS101(x) \wedge TA(x)]$

感觉结论不对，试找反例

结论的非： $\neg \exists x[CS101(x) \wedge TA(x)]$

等价于 $\forall x[\neg CS101(x) \vee \neg TA(x)]$

只要找到一个同学集及其映射，
满足论据，且使得每个成员 x

$\neg CS101(x)$

或 $\neg TA(x)$

建模需要严谨性

- 给定下列两个论据，能推出结论是成立的吗？
 - 论据1：所有上过计科导课的同学都精通Go语言。
 - 论据2：某些精通Go语言的同学可成为下一年计科导课助教。
 - 结论：有些上过计科导的同学可成为下一年计科导课助教。
- 严谨地构建谓词表达式、使用推理规则
 - 谓词逻辑往往不关心原初论据本身的正确性。即，假设给定论据是正确的。
 - 定义下列断言记号
 - $CS101(x)$ ：同学 x 上过计科导
 - $MasterGo(x)$ ：同学 x 精通Go语言
 - $TA(x)$ ：同学 x 可成为下一年计科导课助教
- 结论不对，找简单反例
- 设某年级只有两名同学 a, b ，其映射如下
 - 同学 a ： $CS101(a) = MasterGo(a) = True, TA(a) = False$
 - 同学 b ： $CS101(b) = False, MasterGo(b) = TA(b) = True$
 - 容易验证论据1和2都成立，但结论不成立

错误的建模

$\forall x[MasterGo(x)]$

$\exists x[MasterGo(x) \rightarrow TA(x)]$

$\exists x[CS101(x) \rightarrow TA(x)]$

论据1： $\forall x[CS101(x) \rightarrow MasterGo(x)]$

论据2： $\exists x[MasterGo(x) \wedge TA(x)]$

结论： $\exists x[CS101(x) \wedge TA(x)]$

感觉结论不对，试找反例

结论的非： $\neg \exists x[CS101(x) \wedge TA(x)]$

等价于 $\forall x[\neg CS101(x) \vee \neg TA(x)]$

只要找到一个同学集及其映射，
满足论据，且使得每个成员 x

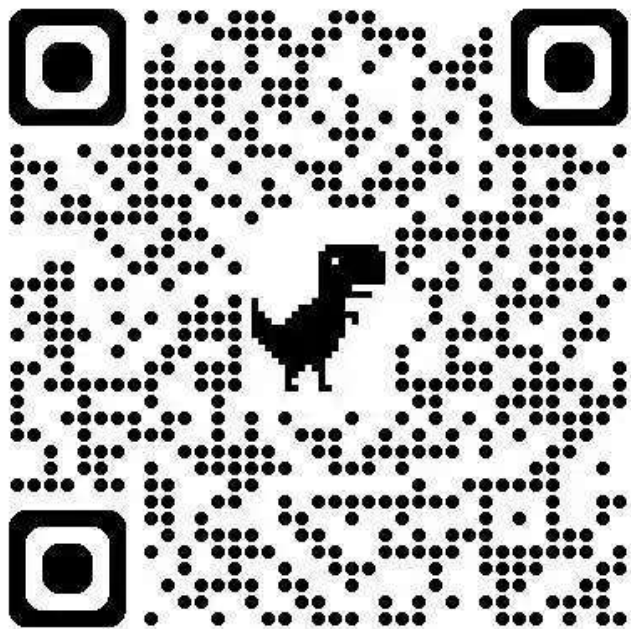
$\neg CS101(x)$

或 $\neg TA(x)$

$\neg TA(a)$

$\neg CS101(b)$

课堂小测验



3.3 图灵机的通用性和局限性

- 图灵机能够求解任意可计算问题
 - 自动写出可计算数的任意长的数字，如
 - $\pi = 3.1415926535897932384626433832795028841971693993751058209749445923078164$
- 邱奇-图灵论题（Church-Turing Hypothesis）
 - 任意抽象计算机与图灵机等价
 - （足够大存储器的）冯诺依曼计算机与图灵机等价
- 存在不可计算问题
 - 例如：停机问题
 - 图灵可计算数是可数集，远远少于实数集（不可数的集合）

图灵机的通用性和局限性

- 邱奇-图灵论题（**Church-Turing Hypothesis**）

- 任意抽象计算机与图灵机等价
 - （足够大存储器的）冯诺依曼计算机与图灵机等价

- 此处的“等价”是什么含义？

- 可计算性意义上等价
- 实用意义上不等价
- 冯诺依曼机是桥接模型，图灵机不是
 - 图灵机加法非常繁琐，但Go程序只需要一行： $X = 1024 + 512$

哥德尔不完备定理的一个实例

- 哥德尔不完备定理：在任何一个包含初等算术的数学系统中，存在真但不可证的命题。
 - 初等算术=皮亚诺算术
 - 哪个真命题在哪个皮亚诺算术系统中不可证？
- Goodstein Theorem, 1944
 - Every Goodstein sequence approaches zero
- 戈德斯坦定理不能在皮亚诺算术系统中证明
 - Kirby & Paris, 1982
 - Cichon, 1983
 - Caicedo, 2007
 - Zankl et al., 2014

皮亚诺算术五公理 (引自大不列颠百科全书)

1. Zero is a natural number.
零是自然数
2. Every natural number has a successor in the natural numbers.
任意自然数有一后继自然数
3. Zero is not the successor of any natural number.
零不是任何自然数的后继
4. If the successor of two natural numbers is the same, then the two original numbers are the same.
后继相同的两个自然数相等
5. If a set contains zero and the successor of every number is in the set, then the set contains the natural numbers. 归纳公理

Goodstein's Theorem

- For any natural number n , the **Goodstein sequence** is $(n)_0, (n)_1, \dots, (n)_{G(n)}, 0, \dots$
- E.g., for $n=266$, the Goodstein sequence is

$$\begin{aligned}
 (266)_0 &= 2^{2^{2+1}} + 2^{2+1} + 2 &= 266 \\
 (266)_1 &= 3^{3^{3+1}} + 3^{3+1} + 3 - 1 &\approx 4.4 \times 10^{38} \\
 (266)_2 &= 4^{4^{4+1}} + 4^{4+1} + 2 - 1 &\approx 3.2 \times 10^{616} \\
 (266)_3 &= 5^{5^{5+1}} + 5^{5+1} + 1 - 1 &\approx 2.5 \times 10^{10921} \\
 (266)_4 &= 6^{6^{6+1}} + 6^{6+1} - 1 &\approx 3.5 \times 10^{217832}
 \end{aligned}$$

.....

整个宇宙基本粒子数 $< 10^{90}$

- 看起来增长很快。但是
 - Goodstein's Theorem:** For any natural number n , its Goodstein sequence $(n)_0, (n)_1, \dots, (n)_{G(n)}, \dots$ approaches 0.
- $G(n)$** 称为 Goodstein function. The first k such that $(n)_k = 0$.

Goodstein sequence for n=4

- 2为底: $(4)_0 = 2^2 = 4$
- 3为底: $(4)_1 = 3^3 - 1 = 2 \cdot 3^2 + 2 \cdot 3 + 2 = 26$
- 4为底: $(4)_2 = 2 \cdot 4^2 + 2 \cdot 4 + 2 - 1 = 41$
- 5为底: $(4)_3 = 2 \cdot 5^2 + 2 \cdot 5 + 1 - 1 = 60$
- 6为底: $(4)_4 = 2 \cdot 6^2 + 2 \cdot 6 - 1 = 83$
- 7为底: $(4)_5 = 2 \cdot 7^2 + 7 + 5 - 1 = 109$
- 8为底: $(4)_6 = 2 \cdot 8^2 + 8 + 4 - 1 = 139$

.....

- $(4)_{G(4)} = 0$
 - $G(4) = 3.2^{402653211} - 2 \approx 6.895 \times 10^{121210694}$

整个宇宙基本粒子数 $< 10^{90}$

- 能够枚举所有 $(4)_k$ 吗?

英文教科书勘误 (p. 121)

- We discuss a specific statement to make our understanding of Gödel's first incompleteness theorem more concrete. The statement is Goodstein theorem, which is true, but cannot be proven in **any** mathematical system that includes elementary number theory.

“any” 应为 “a”

- We discuss a specific statement to make our understanding of Gödel's first incompleteness theorem more concrete. The statement is Goodstein theorem, which is true, but cannot be proven in **a** mathematical system that includes elementary number theory.

4. 算法思维

Acu-Exams-CP

- 强调构造性
- 强调算法复杂度
- 算法思维体现四个巧妙
- 掌握高德纳算法定义
 - 加法图灵机设计是算法设计，状态转移表满足高德纳算法定义
- 理解渐近记号

算法思维妙在何处？

- 算法概念的巧妙定义（高德纳五点定义）
 - 简洁、通用、可检查、可分析
 - 加法图灵机设计是算法设计
- 巧妙的效率度量（复杂度的 o ， O ， Ω ， Θ 记号表达）
 - 忽略不重要细节，简化算法分析
 - 快速排序：最坏复杂度 $O(n^2)$ ，平均复杂度 $O(n \log n)$
- 巧妙的算法设计方法（algorithmic paradigms）
 - 分治、动态规划、贪心、随机
- 巧妙的变奏，适应问题特色
 - 单因素优选法中的0.618法，而不是二分查找法
 - 复用查询结果
 - 二分法的复杂度是 $\log_{\sqrt{2}} n$ ， $> \log_{1.618} n$

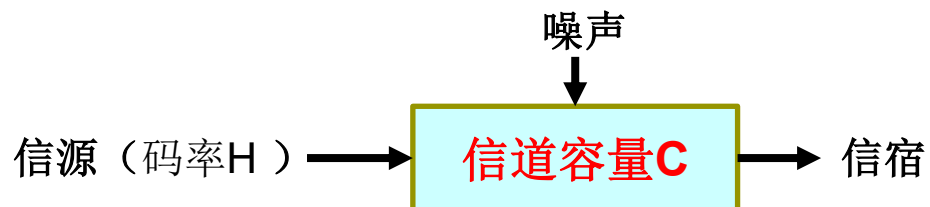
4.1 什么是构造性？

Acu-Exams-CP

- Effectiveness = Constructiveness + Finiteness

(广义) 构造性 = (狭义) 构造性 + 有穷性

- 信息论中的香农第二定理是存在性定理，不是构造性定理



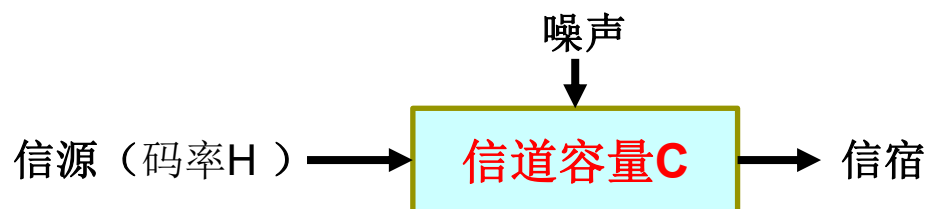
假如 $H \leq C$ ，存在编码，
实现无错消息传递

什么是狭义构造性？

- Effectiveness = Constructiveness + Finiteness

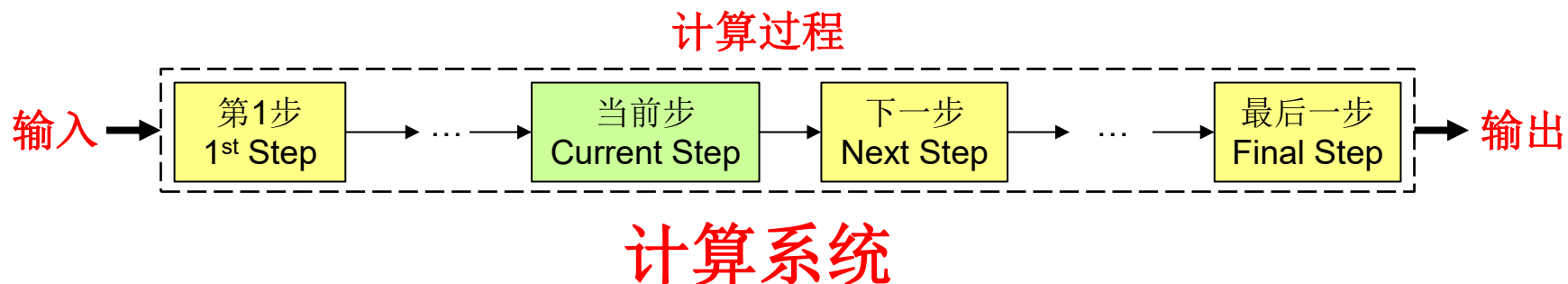
(广义) 构造性 = (狭义) 构造性 + 有穷性

- 信息论中的香农第二定理是存在性定理，不是构造性定理



假如 $H \leq C$ ，存在编码，
实现无错消息传递

- 计算过程是构造性过程，能够一步一步算出（构造出）结果



什么是有穷性？

- Effectiveness = Constructiveness + **Finiteness**

（广义）构造性 = （狭义）构造性 + **有穷性**

- **两类有穷性**

- 需要的资源量与问题规模 N 有关，是 $O(f(N))$ ；
- 需要的资源量与问题规模 N 无关，是 $O(1)$
- 其中，问题规模 N 可以任意大，但不是无穷大

什么是有穷性？

- Effectiveness = Constructiveness + **Finiteness**

(广义) 构造性 = (狭义) 构造性 + **有穷性**

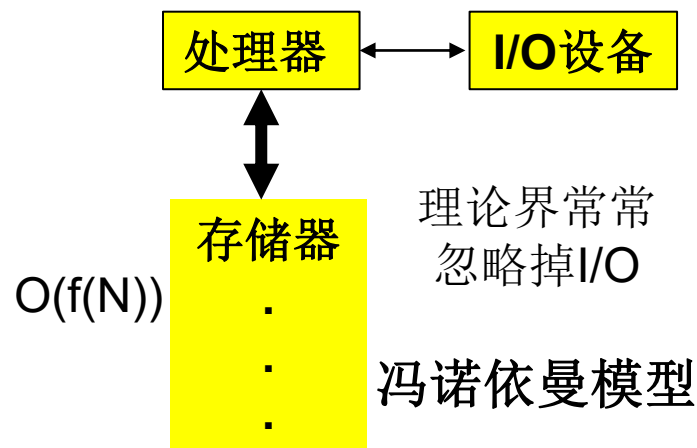
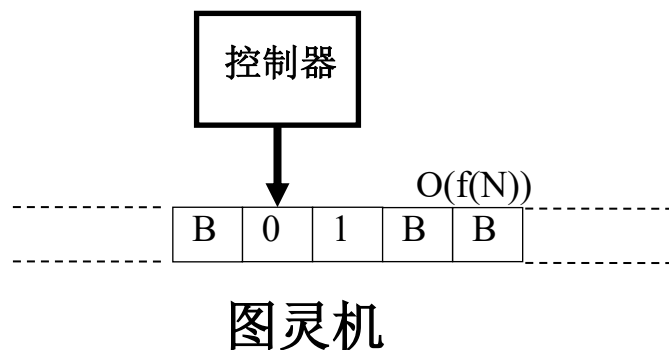
- 两类有穷性：与问题规模 N 有关 $O(f(N))$ ；与问题规模 N 无关 $O(1)$

- **存储有穷性**：与问题规模有关，空间复杂度
 - 笔记本电脑，与图灵机不等价，因为**有限存储容量**
 - (足够大存储容量的) 笔记本电脑，与图灵机等价

$O(f(N))$, N 任意大

$O(1)$

$O(f(N))$



什么是有穷性？

- Effectiveness = Constructiveness + **Finiteness**

(广义) 构造性 = (狭义) 构造性 + **有穷性**

- 两类有穷性：与问题规模 N 有关 $O(f(N))$ ；与问题规模 N 无关 $O(1)$

- **存储有穷性**：与问题规模有关，空间复杂度

- 笔记本电脑，与图灵机不等价，因为**有限存储容量**
- (足够大存储容量的) 笔记本电脑，与图灵机等价

- **程序有穷性**：程序行数与问题规模无关

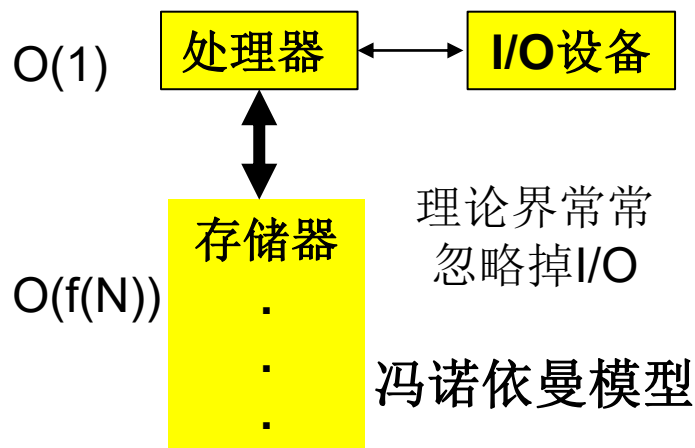
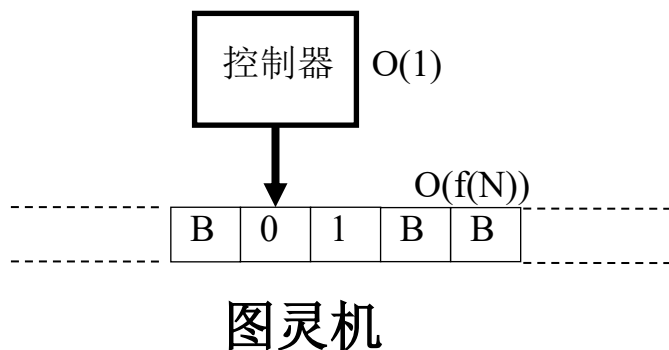
- 用有限行程序表达计算过程
 - Ada运算流程→控制流程序
- 任何图灵机的状态转移表只有有限行
 - 加法图灵机：几十行

$O(f(N))$, N 任意大

$O(1)$

$O(f(N))$

$O(1)$



4.2 掌握高德纳的算法定义

一个算法是一组有限条规则，给出求解特定类型问题的操作序列，并具备下列五个特征：

- ① 有限性：算法在有限步骤之后必然要终止。 $\sim n^2$
- ② 确定性：算法的每个步骤都必须精确地（严格地和无歧义地）定义。
- ③ 输入：算法有零个或多个输入，在算法开始或中途动态地给定。
- ④ 输出：算法有一个或多个输出，输出是与输入有特定关系的数值。
- ⑤ 能行性：算法的所有操作必须是充分基本的，原则上一个人能够用笔和纸在有限时间内精确地完成它们。

简洁

通用

功能可检查

性能可分析

冒泡排序算法

- 输入：待排序的数组 A ，数组 A 的长度 n 。
- 输出：排好序的数组 A 。
- 算法步骤描述：
 for $i = 1$ to $n - 1$
 for $j = 1$ to $n - i$
 if $A[j] > A[j+1]$
 exchange $A[j]$ with $A[j+1]$.

掌握高德纳的算法定义

● 什么是/不是算法

- 一个算法是一组有限条规则，给出求解特定类型问题的操作序列，并具备下列五个特征：
 - ① 有限性：算法在有限步骤之后必然要终止。
 - ② 确定性：算法的每个步骤都必须精确地（严格地和无歧义地）定义。
 - ③ 输入：算法有零个或多个输入，在算法开始或中途动态地给定。
 - ④ 输出：算法有一个或多个输出，输出是与输入有特定关系的数值。
 - ⑤ 能行性：算法的所有操作必须是充分基本的，原则上一个人能够用笔和纸在有限时间内精确地完成它们。

冒泡排序算法

- 输入：待排序的数组 A ，数组 A 的长度 n 。
- 输出：排好序的数组 A 。
- 算法步骤描述：

```
for  $i = 1$  to  $n - 1$ 
  for  $j = 1$  to  $n - i$ 
    if  $A[j] > A[j+1]$ 
      exchange  $A[j]$  with  $A[j+1]$ .
```

什么不是算法？

重新审视高德纳的算法定义

● 注意三种有穷性

一个算法是一组**有限**条规则，给出求解特定类型问题的操作序列，并具备下列五个特征：

- ① 有限性：算法在**有限**步骤之后必然要终止。
- ② 确定性：算法的每个步骤都必须精确地（严格地和无歧义地）定义。
- ③ 输入：算法有零个或多个输入，在算法开始或中途动态地给定。
- ④ 输出：算法有一个或多个输出，输出是与输入有特定关系的数值。
- ⑤ 能行性：算法的所有操作必须是充分基本的，原则上一个人能够用笔和纸在**有限**时间内精确地完成它们。

冒泡排序算法

- 输入：待排序的数组 A ，数组 A 的长度 n 。
- 输出：排好序的数组 A 。
- 算法步骤描述：

```
for  $i = 1$  to  $n - 1$ 
  for  $j = 1$  to  $n - i$ 
    if  $A[j] > A[j+1]$ 
      exchange  $A[j]$  with  $A[j+1]$ .
```

什么不是算法？

重新审视高德纳的算法定义

● 注意三种有穷性

一个算法是一组**有限**条规则，给出求解特定类型问题的操作序列，并具备下列五个特征：

- ① 有限性：算法在**有限**步骤之后必然要终止。
- ② 确定性：算法的每个步骤都必须精确地（严格地和无歧义地）定义。
- ③ 输入：算法有零个或多个输入，在算法开始或中途动态地给定。
- ④ 输出：算法有一个或多个输出，输出是与输入有特定关系的数值。
- ⑤ 能行性：算法的所有操作必须是充分基本的，原则上一个人能够用笔和纸在**有限**时间内精确地完成它们。

冒泡排序算法

- 输入：待排序的数组 A ，数组 A 的长度 n 。
- 输出：排好序的数组 A 。
- 算法步骤描述：
for $i = 1$ to $n - 1$
 for $j = 1$ to $n - i$
 if $A[j] > A[j+1]$
 exchange $A[j]$ with $A[j+1]$.

什么不是算法？

- Ada的运算流程图
 - 程序行数应与问题规模无关

重新审视高德纳的算法定义

● 注意三种有穷性

一个算法是一组**有限**条规则，给出求解特定类型问题的操作序列，并具备下列五个特征：

- ① 有限性：算法在**有限**步骤之后必然要终止。
- ② 确定性：算法的每个步骤都必须精确地（严格地和无歧义地）定义。
- ③ 输入：算法有零个或多个输入，在算法开始或中途动态地给定。
- ④ 输出：算法有一个或多个输出，输出是与输入有特定关系的数值。
- ⑤ 能行性：算法的所有操作必须是充分基本的，原则上一个人能够用笔和纸在**有限**时间内精确地完成它们。

冒泡排序算法

- 输入：待排序的数组 A ，数组 A 的长度 n 。
- 输出：排好序的数组 A 。
- 算法步骤描述：
for $i = 1$ to $n - 1$
 for $j = 1$ to $n - i$
 if $A[j] > A[j+1]$
 exchange $A[j]$ with $A[j+1]$.

什么不是算法？

- Ada的运算流程图
 - 程序行数应与问题规模无关
- 无穷循环不停机

重新审视高德纳的算法定义

● 注意三种有穷性

一个算法是一组**有限**条规则，给出求解特定类型问题的操作序列，并具备下列五个特征：

- ① 有限性：算法在**有限**步骤之后必然要终止。
- ② 确定性：算法的每个步骤都必须精确地（严格地和无歧义地）定义。
- ③ 输入：算法有零个或多个输入，在算法开始或中途动态地给定。
- ④ 输出：算法有一个或多个输出，输出是与输入有特定关系的数值。
- ⑤ 能行性：算法的所有操作必须是充分基本的，原则上一个人能够用笔和纸在**有限**时间内精确地完成它们。

冒泡排序算法

- 输入：待排序的数组 A ，数组 A 的长度 n 。
- 输出：排好序的数组 A 。
- 算法步骤描述：
for $i = 1$ to $n - 1$
 for $j = 1$ to $n - i$
 if $A[j] > A[j+1]$
 exchange $A[j]$ with $A[j+1]$.

什么不是算法？

- Ada的运算流程图
 - 程序行数应与问题规模无关
- 无穷循环不停机
- 包含下列操作
 - if **哥德巴赫猜想为真** {...}
 有限时间完不成！

重新审视高德纳的算法定义

● 注意三种有穷性

一个算法是一组**有限**条规则，给出求解特定类型问题的操作序列，并具备下列五个特征：

- ① 有限性：算法在**有限**步骤之后必然要终止。
- ② 确定性：算法的每个步骤都必须精确地（严格地和无歧义地）定义。
- ③ 输入：算法有零个或多个输入，在算法开始或中途动态地给定。
- ④ 输出：算法有一个或多个输出，输出是与输入有特定关系的数值。
- ⑤ 能行性：算法的所有操作必须是充分基本的，原则上一个人能够用笔和纸在**有限**时间内精确地完成它们。

冒泡排序算法

- 输入：待排序的数组 A ，数组 A 的长度 n 。
- 输出：排好序的数组 A 。
- 算法步骤描述：

```
for  $i = 1$  to  $n - 1$ 
  for  $j = 1$  to  $n - i$ 
    if  $A[j] > A[j+1]$ 
      exchange  $A[j]$  with  $A[j+1]$ .
```

规则有穷性

伪代码行数是 $O(1)$ ，与 n 无关

步骤有穷性

即算法的时间复杂度

操作能行性

任一操作是充分基本的，
所需时间肯定是有限的

重新审视高德纳的算法定义

● 确定性是什么意思？

一个算法是一组有限条规则，给出求解特定类型问题的操作序列，并具备下列五个特征：

- ① 有限性：算法在有限步骤之后必然要终止。
- ② 确定性：算法的每个步骤都必须精确地（严格地和无歧义地）定义。

比特精准！

- ① 输入：算法有零个或多个输入，在算法开始或中途动态地给定。
- ② 输出：算法有一个或多个输出，输出是与输入有特定关系的数值。
- ③ 能行性：算法的所有操作必须是充分基本的，原则上一个人能够用笔和纸在有限时间内精确地完成它们。

冒泡排序算法

- 输入：待排序的数组 A ，数组 A 的长度 n 。
- 输出：排好序的数组 A 。
- 算法步骤描述：

```
for  $i = 1$  to  $n - 1$ 
  for  $j = 1$  to  $n - i$ 
    if  $A[j] > A[j+1]$ 
      exchange  $A[j]$  with  $A[j+1]$ .
```

什么不是算法？

- 包含非比特精准的操作步骤

重新审视高德纳的算法定义

● 确定性是什么意思？

一个算法是一组有限条规则，给出求解特定类型问题的操作序列，并具备下列五个特征：

- ① 有限性：算法在有限步骤之后必然要终止。
- ② 确定性：算法的每个步骤都必须精确地（严格地和无歧义地）定义。
- ③ 输入：算法有零个或多个输入，在算法开始或中途动态地给定。
- ④ 输出：算法有一个或多个输出，输出是与输入有特定关系的数值。
- ⑤ 能行性：算法的所有操作必须是充分基本的，原则上一个人能够用笔和纸在有限时间内精确地完成它们。

冒泡排序算法

- 输入：待排序的数组 A ，数组 A 的长度 n 。
- 输出：排好序的数组 A 。
- 算法步骤描述：

```
for  $i = 1$  to  $n - 1$ 
  for  $j = 1$  to  $n - i$ 
    if  $A[j] > A[j+1]$ 
      exchange  $A[j]$  with  $A[j+1]$ .
```

什么不是算法？

- 包含下列操作
- 比较两个物理线段的长度
If 线段A长度==线段B长度 {...}

A
B

4.3 算法的复杂度分析

- 求解斐波那契数列的递归算法GO代码耗时多少？

```
func fibonacci(n int) int {  
    if n == 0 || n == 1 {  
        return n  
    }  
    return fibonacci(n-1) + fibonacci(n-2)  
}
```

- 复杂度分析（常用求解递推式的分析方法）

- $T(n) = T(n-1) + T(n-2) = T(n-2) + T(n-3) + T(n-3) + T(n-4)$
- $2 * T(n-2) < T(n) < 2 * T(n-1)$
- $T(n) < 2 * T(n-1) < 4 * T(n-2) < 8 * T(n-3) \dots < 2^n$
- $T(n) > 2 * T(n-2) > 4 * T(n-4) > 8 * T(n-6) \dots > 2^{n/2}$
- $2^{n/2} < T(n) < 2^n$
- **$T(n) = \Omega(2^{n/2})$, $T(n) = O(2^n)$, 指数复杂度！**

指数复杂度增长非常迅速！

两国战后谈判，A国要求B国赔偿粮食。
在10x10棋盘上放满稻米：

第1格放1粒米，第2格放2粒米，第3格放4粒米，...，第k格放 2^k 粒米。

B国该不该答应？

10x10棋盘

粒	1	2	4	8	16	32			

指数复杂度增长非常迅速！

在10x10棋盘上放满稻米：

第1格放1粒米，第2格放2粒米，第3格放4粒米，...，第k格放 2^k 粒米。

每粒米质量=0.02克（50粒米=1克）

放满前10个格子，棋盘上稻米才20来克

棋盘上放满稻米总共有 $2^{n+1}-1$ 粒米。

放满100个格子，总共有 $2^{101}-1$ 粒米

地球的质量大约是 6×10^{24} 千克。

B国要赔偿的稻米重量相当于
8个多地球！

10x10棋盘

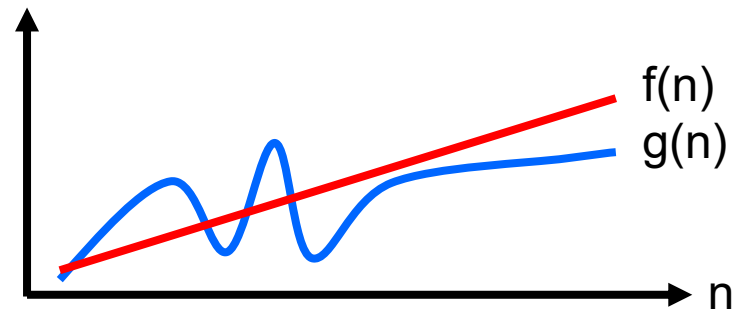
粒	1	2	4	8	16	32	64	128	256	512
克	0.02	0.04	0.08	0.16	0.32	0.64	1.28	2.56	5.12	10.24

小o, 大O, Ω , Θ 记号

等号是单向的, 不同于数学等号

$n^{1.58} = \mathbf{O}(n^2)$, $n^{1.58} = \mathbf{O}(n^3)$, 但 $\mathbf{O}(n^2) \neq \mathbf{O}(n^3)$

假设: 当 n 足够大



- $f(n) = o(g(n))$

- $n^{1.58} = \mathbf{o}(n^2)$, $n^{1.58} \neq \mathbf{o}(n^{1.58})$, $n^2 \neq \mathbf{o}(n^{1.58})$

$$\lim_{n \rightarrow \infty} f(n) / G(n) = 0$$

- $f(n) = O(g(n))$

- $n^{1.58} = \mathbf{O}(n^2)$, $n^{1.58} = \mathbf{O}(n^{1.58})$, $n^2 \neq \mathbf{O}(n^{1.58})$

$$\exists \text{ 常数 } c > 0, f(n) \leq cg(n)$$

- $f(n) = \Omega(g(n))$

- $n^{1.58} \neq \mathbf{\Omega}(n^2)$, $n^{1.58} = \mathbf{\Omega}(n^{1.58})$, $n^2 = \mathbf{\Omega}(n^{1.58})$

$$\exists \text{ 常数 } c > 0, f(n) \geq cg(n)$$

- $f(n) = \Theta(g(n))$

- $n^{1.58} \neq \mathbf{\Theta}(n^2)$, $n^{1.58} = \mathbf{\Theta}(n^{1.58})$, $n^2 \neq \mathbf{\Theta}(n^{1.58})$

$$f(n) = O(g(n)) \text{ 并且 } f(n) = \Omega(g(n))$$

小o, 大O, Ω , Θ 记号

共同假设: 当 n 足够大。 g 是 f 的严格上阶o、上阶O、下阶 Ω 、等阶 Θ

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

● $f(n) = o(g(n))$ g 是 f 的严格上界 (阶)

● $n^{1.58} = o(n^2)$, $n^{1.58} \neq o(n^{1.58})$, $n^2 \neq o(n^{1.58})$

● $f(n) = O(g(n))$ 上界

$$\exists \text{常数 } c > 0, f(n) \leq cg(n)$$

● $n^{1.58} \neq O(n^2)$, $n^{1.58} = O(n^{1.58})$, $n^2 \neq O(n^{1.58})$

● $f(n) = \Omega(g(n))$ 下界

$$\exists \text{常数 } c > 0, f(n) \geq cg(n)$$

● $n^{1.58} \neq \Omega(n^2)$, $n^{1.58} = \Omega(n^{1.58})$, $n^2 = \Omega(n^{1.58})$

● $f(n) = \Theta(g(n))$ 等阶

$$f(n) = O(g(n)) \text{ 并且 } f(n) = \Omega(g(n))$$

● $n^{1.58} \neq \Theta(n^2)$, $n^{1.58} = \Theta(n^{1.58})$, $n^2 \neq \Theta(n^{1.58})$

5. 计算系统思维

- 通过**抽象**，将**模块**组合成为系统，**无缝**执行计算过程
 - 抽象化：一个通用抽象代表多个具体需求
 - 模块化：系统由多个模块组合而成
 - 计算机 = 硬件 + 系统软件 + 应用软件
 - 信息隐藏原理，接口概念
 - 无缝衔接
 - 避免缝隙：
 - 扬雄周期原理、波斯特尔鲁棒性原理、冯诺依曼穷举原理
 - 重视瓶颈：阿姆达尔定律
 - 系统性能改进受限于系统瓶颈（针对某任务）
 - $\text{加速比} = 1 / ((1-f)/p + f) \rightarrow 1/f$ 当 $p \rightarrow \infty$

5.1 以一耦万

①向编程者提供抽象，②比特精准地将抽象映射到硬件

万千应用

程序
进程
指令
冯氏结构
指令流水线
时序电路
组合电路

本课程学到的一些基本抽象

- 是什么、通过实例解释、能够举一反三

Data Type 数据类型	bit (1 bit), hexadecimal number (4 bits), byte (8 bits), uint8 (8-bit unsigned integer), integer (64 bits); array (n elements of the same type), slice (a descriptor pointing to an array); text file, BMP image file; hypertext and hyperlink 比特、字节、整数、数组、切片、文本文件、图像文件、超链接	
Software 软件	Algorithm 算法	Smart method of information transformation, such as quicksort, hiding text in a BMP file, etc. 快排算法
	Program 程序	Code realizing algorithms in computer language, such as hide.go in the Text Hider project 信息隐藏程序 hide.go
	Process 进程	Program in execution, such as the “hide” process running in a Linux environment > ./WebServer & 系统显示 PID=79
	Instruction 指令	The smallest unit of software, directly executable by computer hardware 斐波那契计算机指令集
von Neumann Architecture: a computer model bridging software and hardware 冯诺依曼计算机模型		
Hardware 硬件	Instruction Pipeline 指令流水线	The basic hardware mechanism to automatically execute any instruction “取指-译码-执行” 三级流水线
	Sequential Circuit 时序电路	More precisely, only consider Synchronous Sequential Circuit comprised of combinational circuits and state circuits and driven by a clock signal; equivalent to the automata concept 串行加法器
	Combinational Circuit 组合电路	Also known as Boolean circuit, realizing a Boolean function 使用全加器的波纹进位加法器

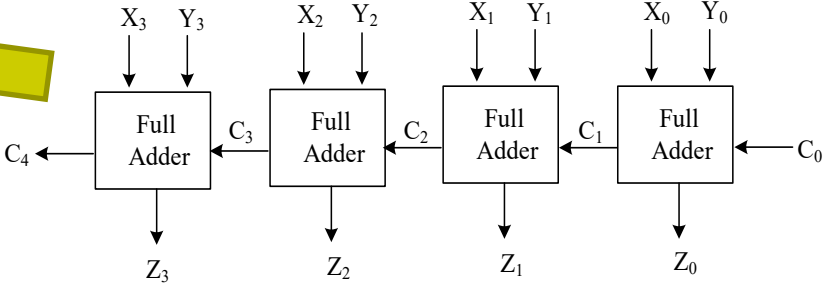
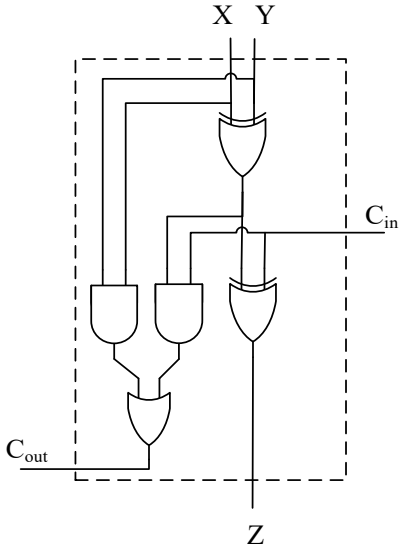
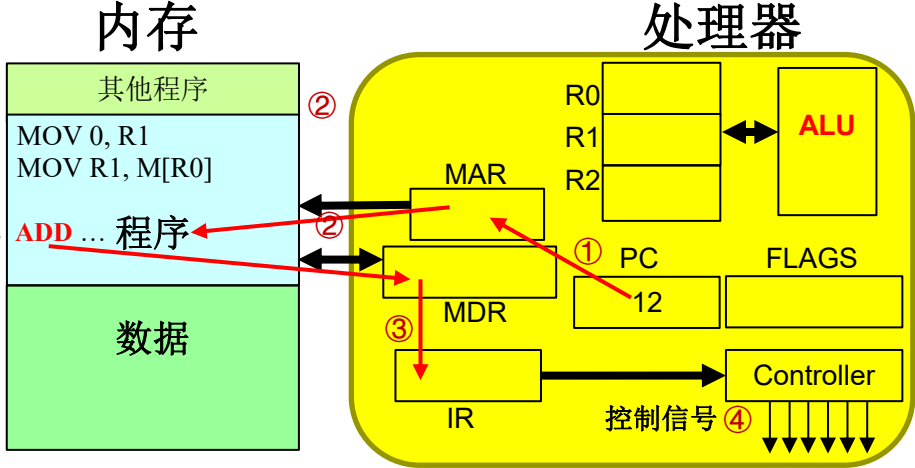
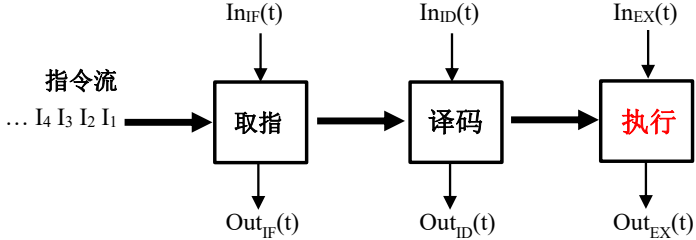
比特精准地将抽象映射到最底层硬件

```
fib[0] = 0
fib[1] = 1
for i := 2; i < 51; i++ {
    fib[i] = fib[i-1] + fib[i-2]
}
```

万千应用

程序
进程
指令
冯氏结构
指令流水线
时序电路
组合电路

```
MOV 0, R1
MOV R1, M[R0]
MOV 1, R1
MOV R1, M[R0+8]
MOV 2, R2 // i:=2
MOV 0, R1 // label Loop
ADD M[R0+R2*8-16], R1
ADD M[R0+R2*8-8], R1
MOV R1, M[R0+R2*8-0]
INC R2 // i++
CMP 51, R2 // i < 51?
JL Loop // if '<' goto Loop
```



信息隐藏程序回顾

- Write a program hide-0.go
 - to hide the text of Shakespeare's *Hamlet*
 - in a picture file Autumn.bmp
 - such that the doctored file shows no visible difference from the original picture
- and another program show-0.go to recover the text

HAMLET

DRAMATIS PERSONAE

CLAUDIUS king of Denmark. (KING CLAUDIUS:)

HAMLET son to the late, and nephew to the present king.

.....

ACT ISCENE I Elsinore. A platform before the castle.

.....

PRINCE FORTINBRAS Let four captains
Bear Hamlet, like a soldier, to the stage;
For he was likely, had he been put on,
To have proved most royally: and, for his passage,
The soldiers' music and the rites of war
Speak loudly for him.
Take up the bodies: such a sight as this
Becomes the field, but here shows much amiss.
Go, bid the soldiers shoot.
[A dead march. Exeunt, bearing off the dead
bodies; after which a peal of ordnance is shot off]

.....

] 换行键0x0A



Original picture

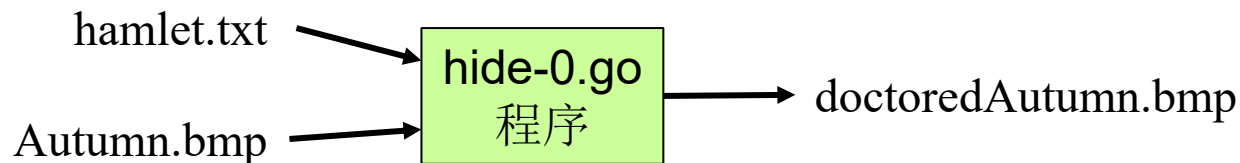


After careless hiding



After careful hiding

程序hide-0.go的设计思路：自顶向下



信息隐藏算法

- 输入：文本文件hamlet.txt 与图像文件 Autumn.bmp。
- 输出：新图像文件doctoredAutumn.bmp，它隐藏了hamlet.txt的全部内容，且与Autumn.bmp无可见差别。
- 计算过程的步骤：
 1. 将文本文件hamlet.txt读进变量t
 2. 将图像文件Autumn.bmp读进变量p
 3. 将hamlet.txt的文件长度隐藏到变量p中Pixel Array的头32个字节
 4. 将hamlet.txt的文件内容隐藏到变量p中Pixel Array的剩余字节
 5. 将p写到新文件doctoredAutumn.bmp

// t 代表文本 (text)
// p 代表图像 (picture)

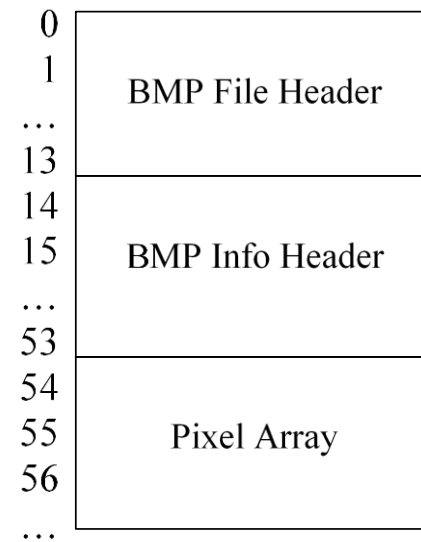
读文件
读文件
隐藏文件长度
隐藏文件内容
写文件

要点1：将文件以字节切片的方式读进内存，以便定位和操作

要点2：使用“基址+索引+偏移量”寻址模式，便于循环



Autumn.bmp



在图像中隐藏文本

- 代码
 - p, _ := ioutil.ReadFile("./Autumn.bmp")
to read the image file into byte slice p
 - t, _ := ioutil.ReadFile("./hamlet.txt")
to read the text file into byte slice t
 - 使用循环语句隐藏t[i], i = 0~len(t)

```
for i:=0; i<len(t); i++){  
    offset := S+T+(i*4)  
    modify(int(t[i]), p[offset:offset+C], C)  
}
```

比例因子是4，因为每个字符有8位，需要p的4个字节。这4个字节的偏移量分别是0、1、2、3。第0次迭代执行 modify(int(t[0]), p[86:90], 4)，修改 P[86]，P[87]，P[88]，P[89]

基址+索引*4+偏移量

回忆支持循环的基址+索引+偏移量寻址模式

- $R1 + M[R0 + R2 * 8 - 16]$
基址 索引 比例因子 偏移量

→ R1



Autumn.bmp
Original p

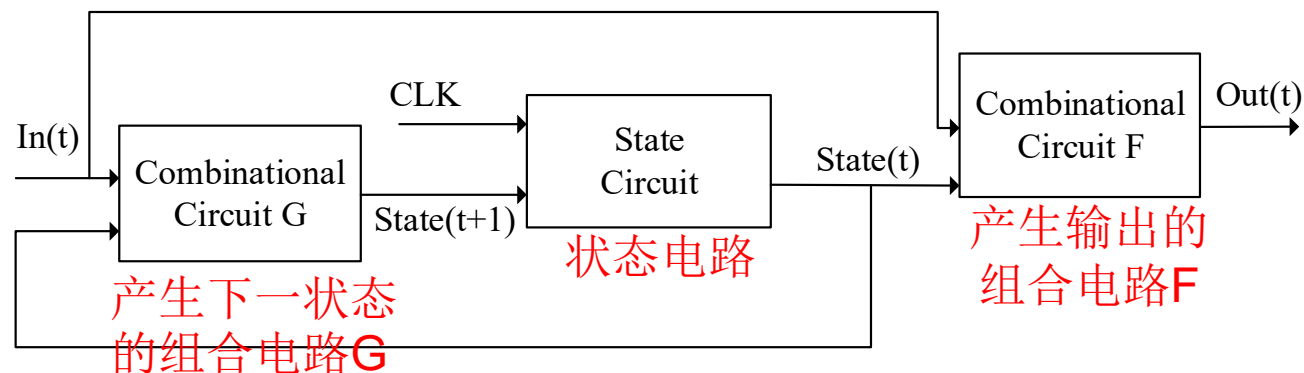
0	BMP FILE HEADER
1	
...	
13	BMP INFO HEADER
14	
15	
...	Pixel Array
53	
54	
55	
56	
...	
85	
86	
87	
88	
89	
90	
91	
92	
93	

doctoredAutumn.bmp
Modified p

0	BMP FILE HEADER
1	
...	
13	BMP INFO HEADER
14	
15	
...	Pixel Array
53	
54	
55	
56	
...	
85	
86	
87	
88	
89	
90	
91	
92	
93	

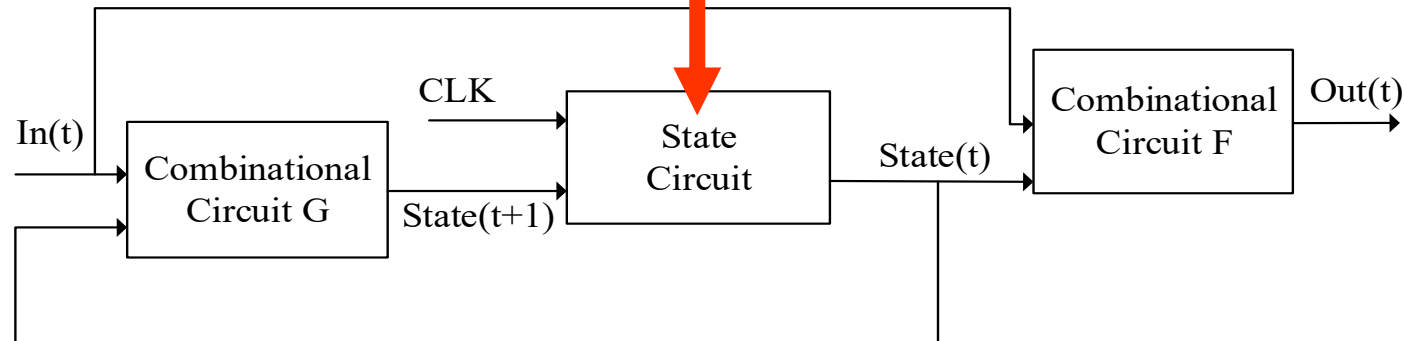
5.2 时序电路

- 两个组合电路，一个状态电路；时钟信号驱动
- 状态电路由一个或多个D触发器组成
- 两个组合电路是
 - 输出电路F: $\text{Out}(t) = F(\text{In}(t), \text{State}(t))$
 - 下一状态电路G: $\text{State}(t+1) = G(\text{In}(t), \text{State}(t))$
- 时序电路的逻辑线路图（时序电路的一般组成）
 - 也称为时钟同步的时序电路（synchronous sequential circuit）



示例：设计4位串行加法器

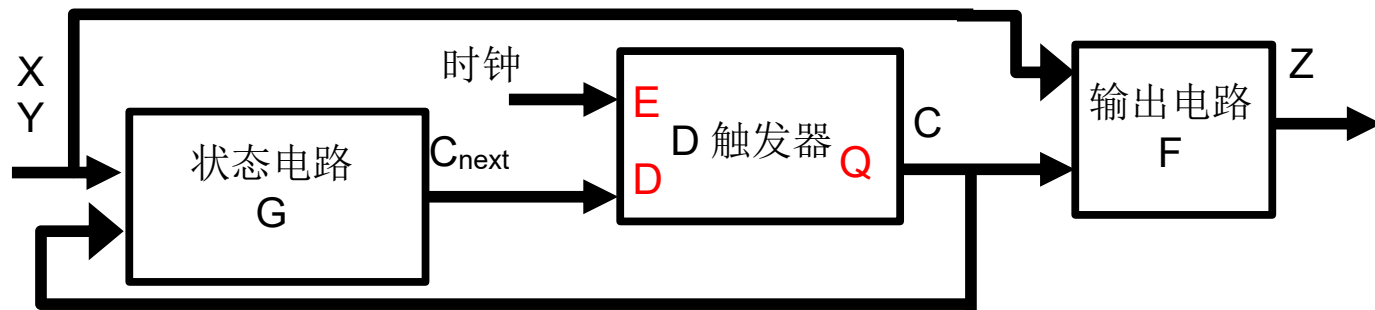
- 在四个时钟周期完成加法 $Z_3Z_2Z_1Z_0 = X_3X_2X_1X_0 + Y_3Y_2Y_1Y_0$
 - 每一周期完成一个全加操作
- 再用一个实例验证
 - $11_{10} + 9_{10} = 1011_2 + 1001_2 = 10100_2 = 20_{10} = 4_{10}$ and overflow
输入 $X=1011$ ， $Y=1001$ ；则输出 $Z=0100$ 且产生进位溢出
- 设计过程
 - 第1步：首先确定状态电路，需要多少个D触发器？
 - 只需要一个 D触发器，其状态Q表示当前进位



设计4位串行加法器

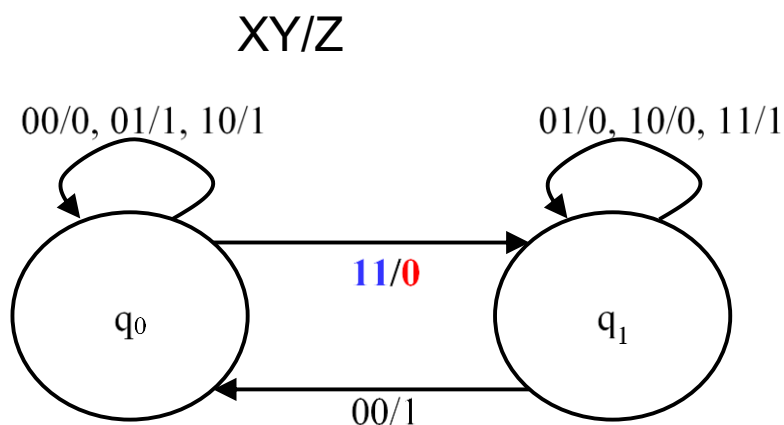
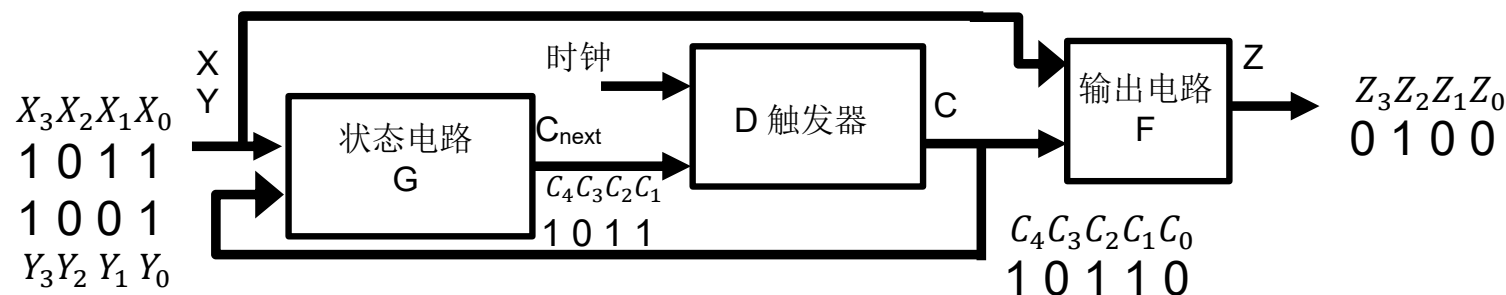
- 设计过程

- 第1步：首先确定状态电路，需要多少个D触发器？
- 只需要一个 D触发器，其状态Q表示当前进位C
- 第2步：套用时序电路一般结构
- In是输入X，Y； Out是输出Z； Q是进位C； Enable=时钟信号CLK



设计过程

- 第3步：求输出电路F与状态电路G的布尔表达式
 - 画出时序电路的状态转换图



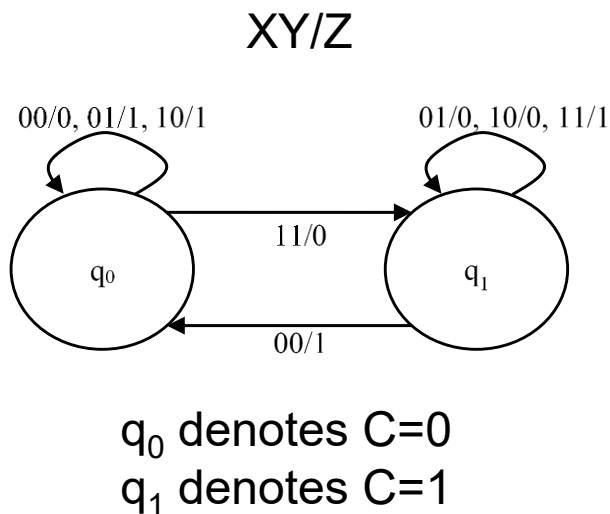
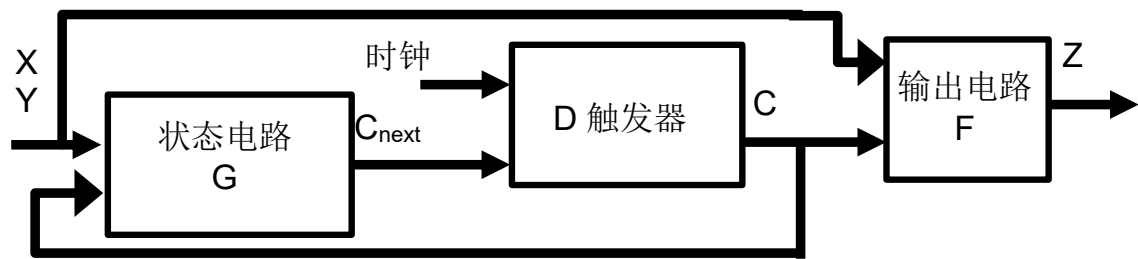
画出时序电路的状态转换图

q_0 denotes $C=0$
 q_1 denotes $C=1$

$$\begin{array}{r}
 101\mathbf{1} = X \\
 100\mathbf{1} = Y \\
 \hline
 101\mathbf{1}\mathbf{0} = C \\
 010\mathbf{0} = Z
 \end{array}$$

设计过程

- 第3步：求输出电路F与状态电路G的布尔表达式
 - 画出时序电路的状态转换图；进而导出真值表

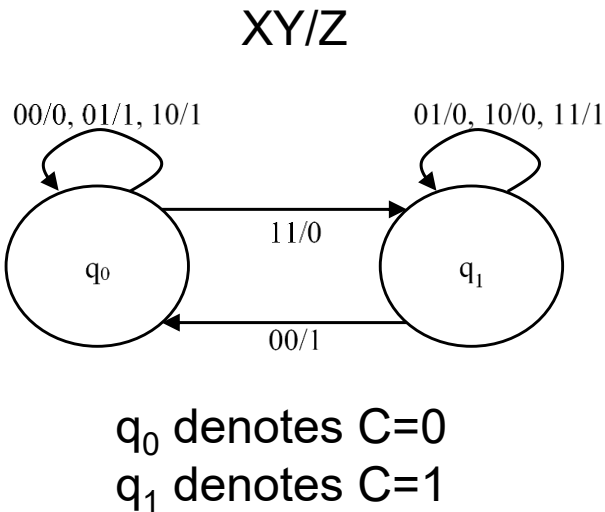
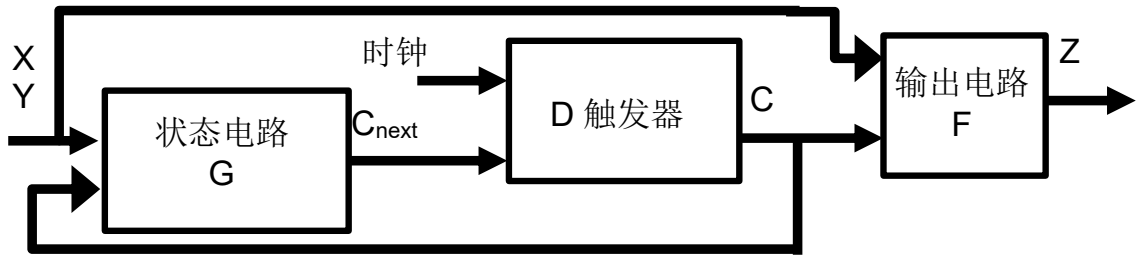


C	X	Y	Z	C_{next}
q_0	0	0	0	q_0
q_0	0	1	1	q_0
q_0	1	0	1	q_0
q_0	1	1	0	q_1
q_1	0	0	1	q_0
q_1	0	1	0	q_1
q_1	1	0	0	q_1
q_1	1	1	1	q_1

C	X	Y	Z	C_{next}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

设计过程

- 第3步：求输出电路F与状态电路G的布尔表达式
 - 画出时序电路的状态转换图；进而导出真值表；进而求出F和G的布尔表达式



C	X	Y	Z	C _{next}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

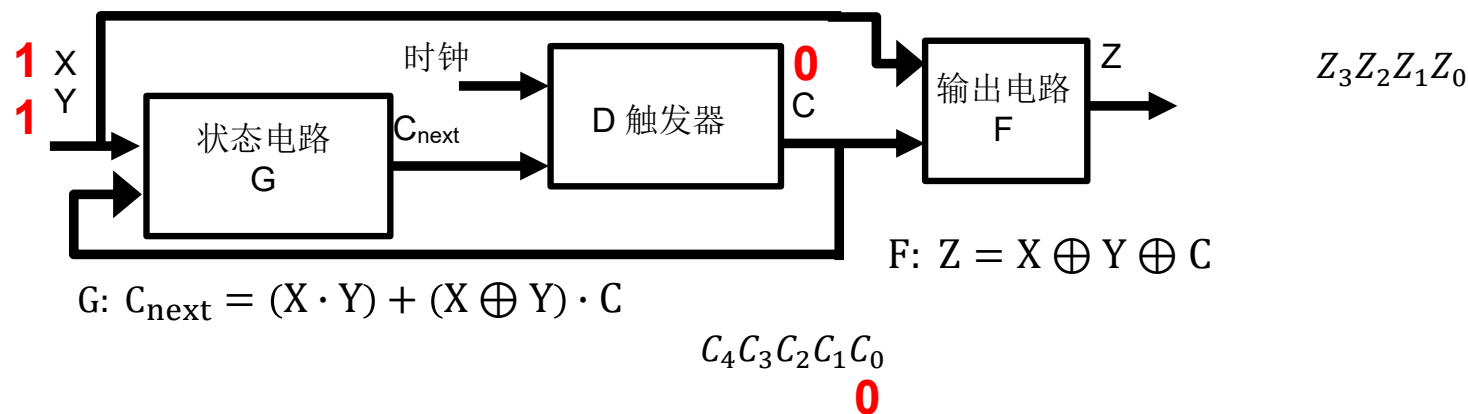
$$Z = F(X, Y, C) = X \oplus Y \oplus C$$

$$C_{next} = G(X, Y, C) = (X \cdot Y) + (X \oplus Y) \cdot C$$

设计过程第4步：验算

- 给定
 - (1) 设计好的下图时序电路,
 - (2) 输入 $X_3X_2X_1X_0 = 1011$, $Y_3Y_2Y_1Y_0 = 1001$, 与初始进位 $C_0 = 0$
 - 正确结果应为 $Z_3Z_2Z_1Z_0 = 0100$, 且有 $C_4 = 1$ 表示溢出
- 验算步骤
 - 时钟周期0: $Z_0 = X_0 \oplus Y_0 \oplus C_0 = 1 \oplus 1 \oplus 0$;
 - $C_1 = (X_0 \cdot Y_0) + (X_0 \oplus Y_0) \cdot C_0 = (1 \cdot 1) + (1 \oplus 1) \cdot 0$

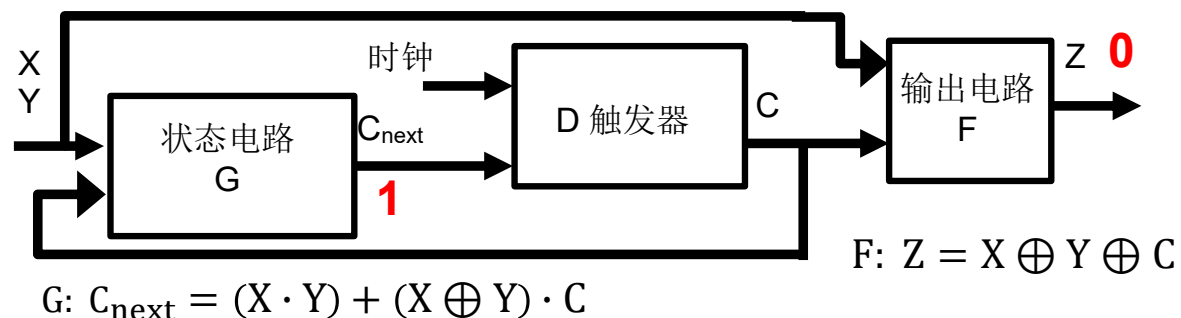
$X_3X_2X_1X_0$
1 0 1 1
 $Y_3Y_2Y_1Y_0$
1 0 0 1



设计过程第4步：验算

- 给定
 - (1) 设计好的下图时序电路,
 - (2) 输入 $X_3X_2X_1X_0 = 1011$, $Y_3Y_2Y_1Y_0 = 1001$, 与初始进位 $C_0 = 0$
 - 正确结果应为 $Z_3Z_2Z_1Z_0 = 0100$, 且有 $C_4 = 1$ 表示溢出
- 验算步骤
 - 时钟周期0: $Z_0 = X_0 \oplus Y_0 \oplus C_0 = 1 \oplus 1 \oplus 0 = 0$;
 - $C_1 = (X_0 \cdot Y_0) + (X_0 \oplus Y_0) \cdot C_0 = (1 \cdot 1) + (1 \oplus 1) \cdot 0 = 1$

$X_3X_2X_1X_0$
1 0 1 1
 $Y_3Y_2Y_1Y_0$
1 0 0 1



$Z_3Z_2Z_1Z_0$
0 1 0 0

$C_4C_3C_2C_1C_0$
1 0 0 0 0

设计过程第4步：验算

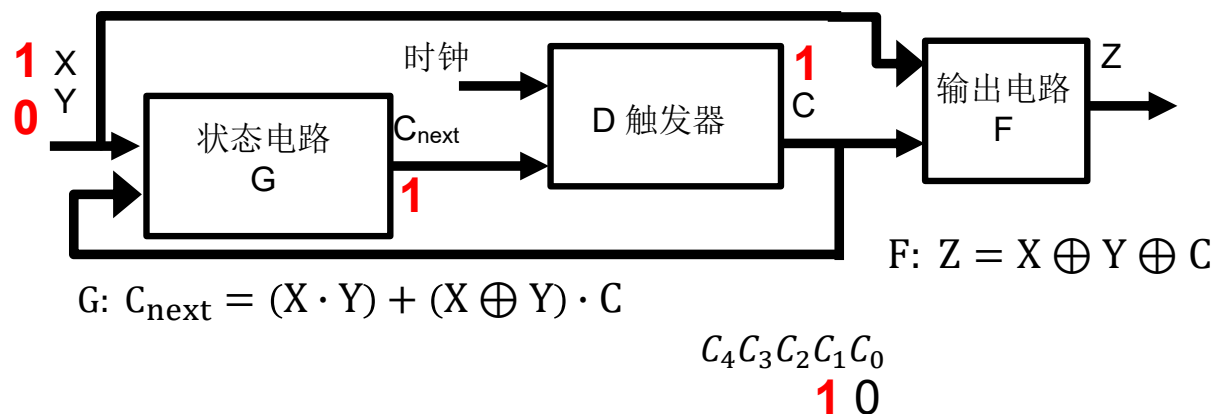
- 给定

- (1) 设计好的下图时序电路，
- (2) 输入 $X_3X_2X_1X_0 = 1011$, $Y_3Y_2Y_1Y_0 = 1001$, 与初始进位 $C_0 = 0$
- 正确结果应为 $Z_3Z_2Z_1Z_0 = 0100$, 且有 $C_4 = 1$ 表示溢出

- 验算步骤

- 时钟周期0: $Z_0 = X_0 \oplus Y_0 \oplus C_0 = 1 \oplus 1 \oplus 0 = 0$; $C_1 = (X_0 \cdot Y_0) + (X_0 \oplus Y_0) \cdot C_0 = (1 \cdot 1) + (1 \oplus 1) \cdot 0 = 1$
- 时钟周期1: $Z_1 = X_1 \oplus Y_1 \oplus C_1 = 1 \oplus 0 \oplus 1$; $C_2 = (X_1 \cdot Y_1) + (X_1 \oplus Y_1) \cdot C_1 = (1 \cdot 0) + (1 \oplus 0) \cdot 1$

$X_3X_2X_1X_0$
1 0 1 1
 $Y_3Y_2Y_1Y_0$
1 0 0 1

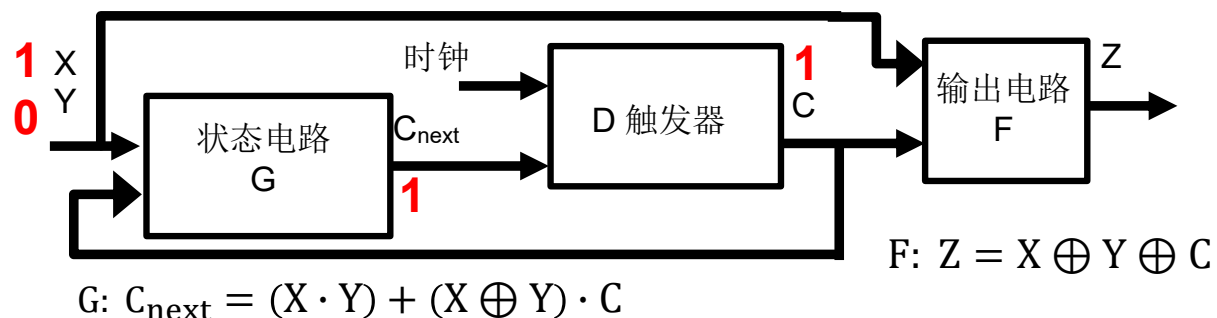


$Z_3Z_2Z_1Z_0$
0

设计过程第4步：验算

- 给定
 - (1) 设计好的下图时序电路,
 - (2) 输入 $X_3X_2X_1X_0 = 1011$, $Y_3Y_2Y_1Y_0 = 1001$, 与初始进位 $C_0 = 0$
 - 正确结果应为 $Z_3Z_2Z_1Z_0 = 0100$, 且有 $C_4 = 1$ 表示溢出
- 验算步骤
 - 时钟周期0: $Z_0 = X_0 \oplus Y_0 \oplus C_0 = 1 \oplus 1 \oplus 0 = 0$; $C_1 = (X_0 \cdot Y_0) + (X_0 \oplus Y_0) \cdot C_0 = (1 \cdot 1) + (1 \oplus 1) \cdot 0 = 1$
 - 时钟周期1: $Z_1 = X_1 \oplus Y_1 \oplus C_1 = 1 \oplus 0 \oplus 1 = 0$; $C_2 = (X_1 \cdot Y_1) + (X_1 \oplus Y_1) \cdot C_1 = (1 \cdot 0) + (1 \oplus 0) \cdot 1 = 1$

$X_3X_2X_1X_0$
1 0 1 1
 $Y_3Y_2Y_1Y_0$
1 0 0 1



$Z_3Z_2Z_1Z_0$
0 1 0 0

$C_4C_3C_2C_1C_0$
1 1 0

设计过程第4步：验算

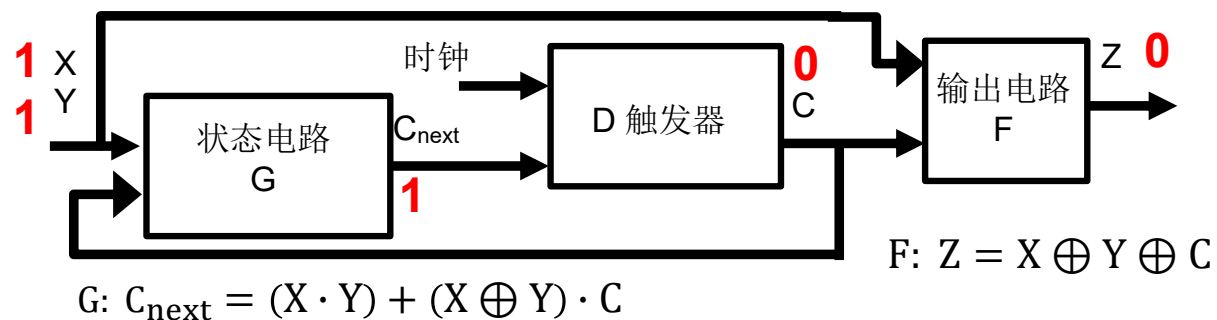
- 给定

- (1) 设计好的下图时序电路,
- (2) 输入 $X_3X_2X_1X_0 = 1011$, $Y_3Y_2Y_1Y_0 = 1001$, 与初始进位 $C_0 = 0$
- 正确结果应为 $Z_3Z_2Z_1Z_0 = 0100$, 且有 $C_4 = 1$ 表示溢出

- 验算步骤

- 时钟周期0: $Z_0 = X_0 \oplus Y_0 \oplus C_0 = 1 \oplus 1 \oplus 0 = 0$; $C_1 = (X_0 \cdot Y_0) + (X_0 \oplus Y_0) \cdot C_0 = (1 \cdot 1) + (1 \oplus 1) \cdot 0 = 1$
- 时钟周期1: $Z_1 = X_1 \oplus Y_1 \oplus C_1 = 0 \oplus 0 \oplus 1 = 1$; $C_2 = (X_1 \cdot Y_1) + (X_1 \oplus Y_1) \cdot C_1 = (0 \cdot 0) + (0 \oplus 0) \cdot 1 = 0$
- 时钟周期2: $Z_2 = X_2 \oplus Y_2 \oplus C_2 = 1 \oplus 0 \oplus 0 = 1$; $C_3 = (X_2 \cdot Y_2) + (X_2 \oplus Y_2) \cdot C_2 = (1 \cdot 0) + (1 \oplus 0) \cdot 0 = 0$
- 时钟周期3: $Z_3 = X_3 \oplus Y_3 \oplus C_3 = 1 \oplus 1 \oplus 0 = 0$; $C_4 = (X_3 \cdot Y_3) + (X_3 \oplus Y_3) \cdot C_3 = (1 \cdot 1) + (1 \oplus 1) \cdot 0 = 1$

$X_3X_2X_1X_0$
1 0 1 1
1 0 0 1
 $Y_3Y_2Y_1Y_0$



$Z_3Z_2Z_1Z_0$
0 1 0 0

$C_4C_3C_2C_1C_0$
1 0 1 1 0

设计过程第4步：验算

- 给定
 - (1) 设计好的下图时序电路,
 - (2) 输入 $X_3X_2X_1X_0 = 1011$, $Y_3Y_2Y_1Y_0 = 1001$, 与初始进位 $C_0 = 0$
 - 正确结果应为 $Z_3Z_2Z_1Z_0 = 0100$, 且有 $C_4 = 1$ 表示溢出
- 验算步骤
 - 时钟周期0: $Z_0 = X_0 \oplus Y_0 \oplus C_0 = 1 \oplus 1 \oplus 0 = 0$; $C_1 = (X_0 \cdot Y_0) + (X_0 \oplus Y_0) \cdot C_0 = (1 \cdot 1) + (1 \oplus 1) \cdot 0 = 1$
 - 时钟周期1: $Z_1 = X_1 \oplus Y_1 \oplus C_1 = 1 \oplus 0 \oplus 1 = 0$; $C_2 = (X_1 \cdot Y_1) + (X_1 \oplus Y_1) \cdot C_1 = (1 \cdot 0) + (1 \oplus 0) \cdot 1 = 1$
 - 时钟周期2: $Z_2 = X_2 \oplus Y_2 \oplus C_2 = 0 \oplus 0 \oplus 1 = 1$; $C_3 = (X_2 \cdot Y_2) + (X_2 \oplus Y_2) \cdot C_2 = (0 \cdot 0) + (0 \oplus 0) \cdot 1 = 0$
 - 时钟周期3: $Z_3 = X_3 \oplus Y_3 \oplus C_3 = 1 \oplus 1 \oplus 0 = 0$; $C_4 = (X_3 \cdot Y_3) + (X_3 \oplus Y_3) \cdot C_3 = (1 \cdot 1) + (1 \oplus 1) \cdot 0 = 1$

举一反三：用时序电路实现4位串行减法器

- 在四个时钟周期完成减法 $Z_3Z_2Z_1Z_0 = X_3X_2X_1X_0 - Y_3Y_2Y_1Y_0$
 - 假设数据表示是二进制补码
 - 四个时钟周期后得到正确结果
- 再用一个实例验证
 - $(-5) - 1 = -6$
- 课上已经讨论过“加减器”

加減器

代表了设计方法4：使用多路复用器和控制信号（本例中的选择输入S）

- 采用多路复用器（multiplexer, MUX）的加減器

- 控制信号S选择做加法(S=0)还是做減法(S=1)

- 采用二进制补码，減法等价于加负数

- X的负数(-X)刚好是X的二进制补码

$$5-5 = 5+(-5); \quad \text{減5 等价于 加-5}$$

- 假设 $X = Y = 5$ 。即， $X_3X_2X_1X_0 = Y_3Y_2Y_1Y_0 = 0101$

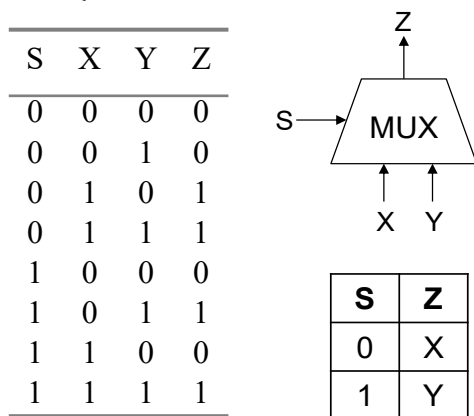
$$X - Y = 5 + (-5) = 5 + (5 \text{ 的补码})$$

$$\rightarrow 0101 + (\overline{0101} + 0001)$$

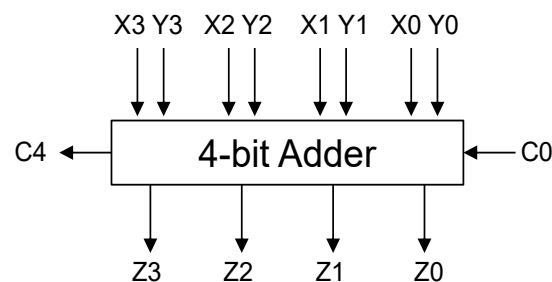
$$= 0101 + 1011$$

$$= 10000 = C_4 Z_3 Z_2 Z_1 Z_0$$

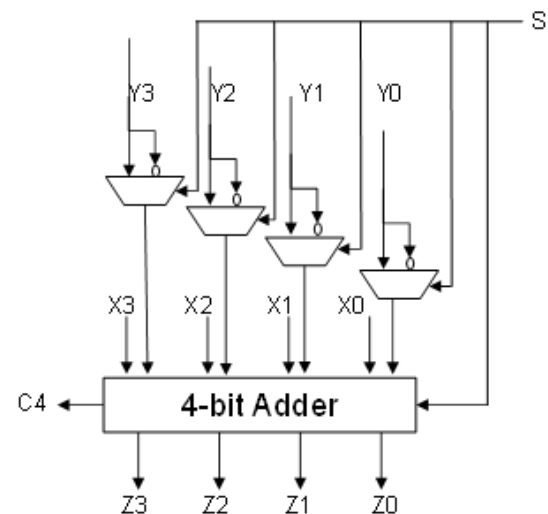
$$Z = \begin{cases} X & \text{if } S = 0 \\ Y & \text{if } S = 1 \end{cases}$$



Multiplexer 多路复用器



Two's complement 4-bit adder



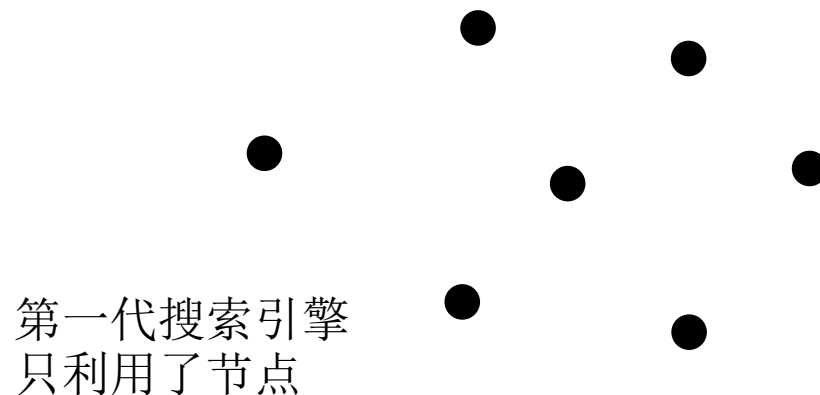
Adder-subtractor

6. 网络思维

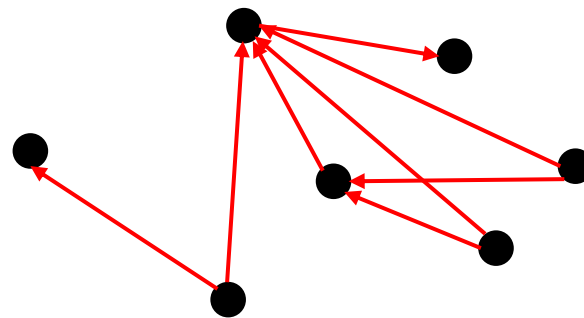
- 通过连通性与协议栈，研究多个节点连接而成的网络
 - 很多问题涉及用户/数据/算法/部件的连接体，而非单体
- 连接体就是网络，即多个节点连接或通信的整体
 - 一个网络的连通性往往可用网络拓扑（一个图）表示
- 协议栈用于节点间通信

6.1 网络拓扑：第二代搜索引擎示例

- 网络拓扑思想：一个网络是一个图，包含两个集合：节点集合与边集合
- 第一代搜索引擎只利用了网页网络的节点集合（网页）
 - Computed search results by matching the keywords in search queries to the contents of webpages (**nodes**)
- 第二代搜索引擎利用了网络拓扑，即网页网络的节点集合（网页）与边集合（超链接）
 - Web links (**edges**) also significantly influence the relevance of search results



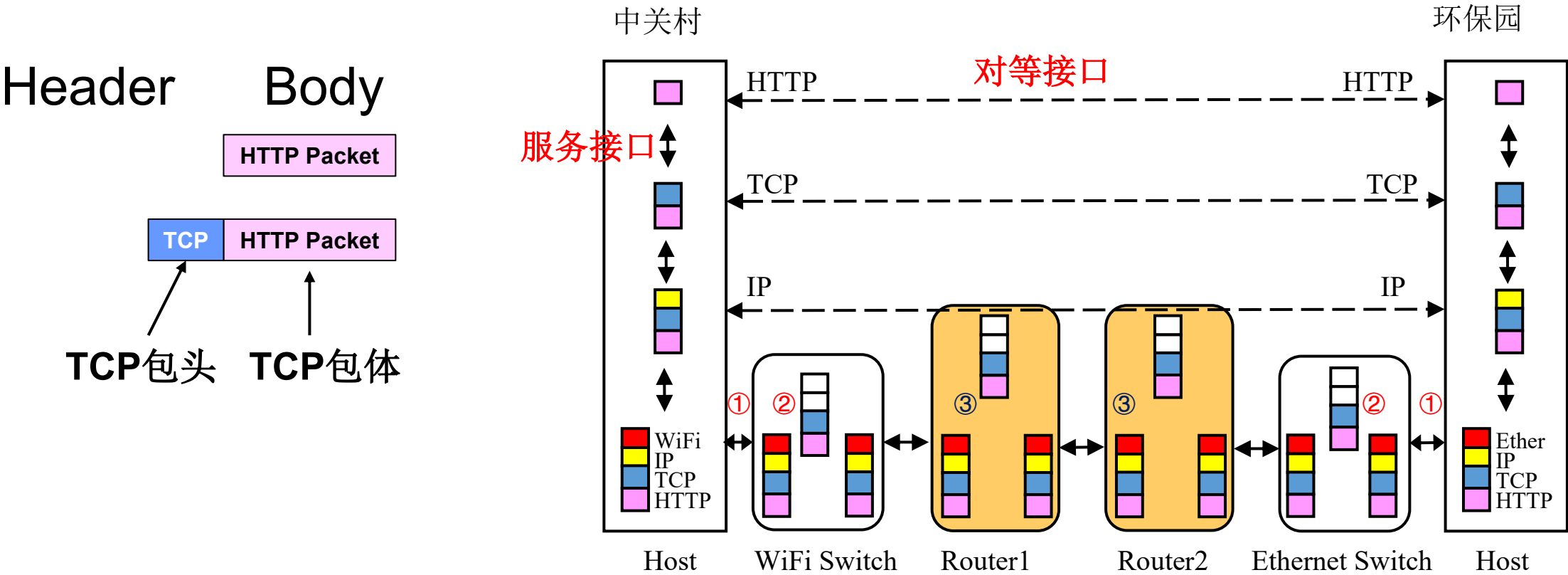
第二代搜索引擎
利用了节点和边



6.2 协议栈：互联网通信示例



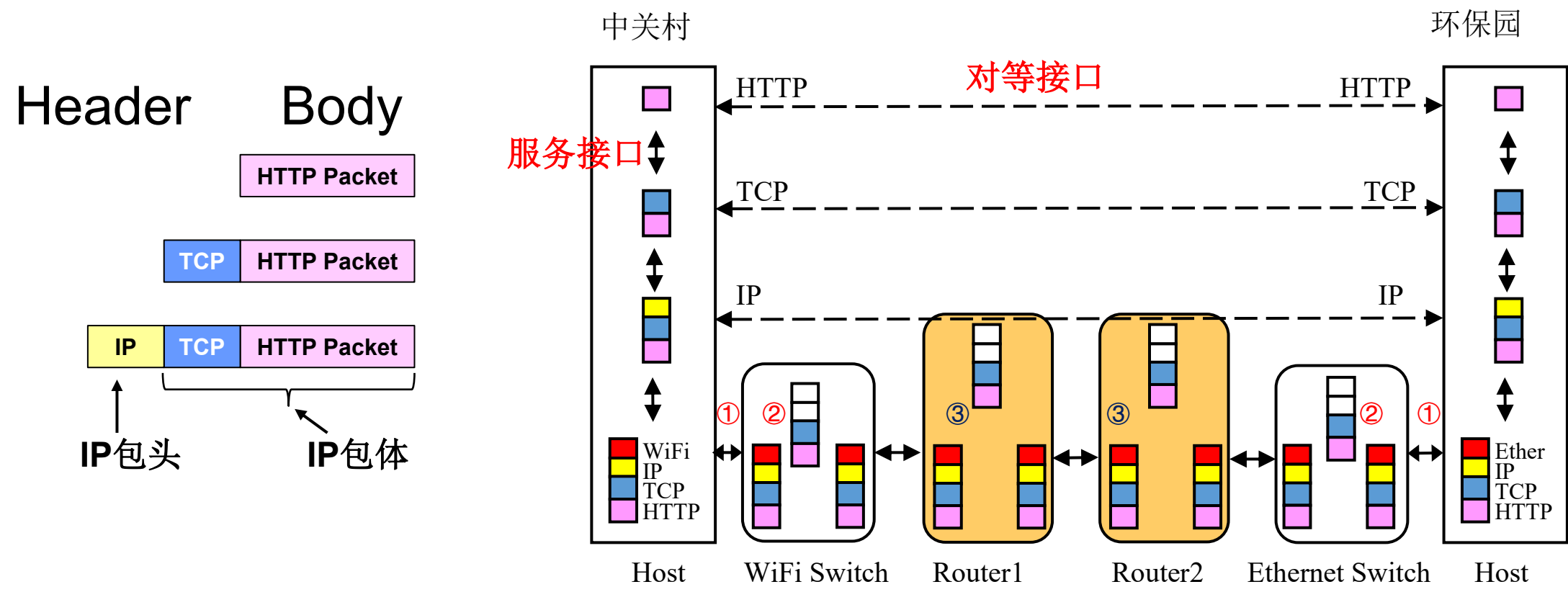
- 从中关村计算所的笔记本电脑通过WiFi访问位于环保园服务器的“猫猫指挥家”图片
- 笔记本电脑IP为192.168.1.101 ， 服务器IP为10.208.104.3， 图片URL为：
http://seafie.solid.things.ac.cn:7019/kitty_band/*0555.JPG



协议栈：互联网通信示例



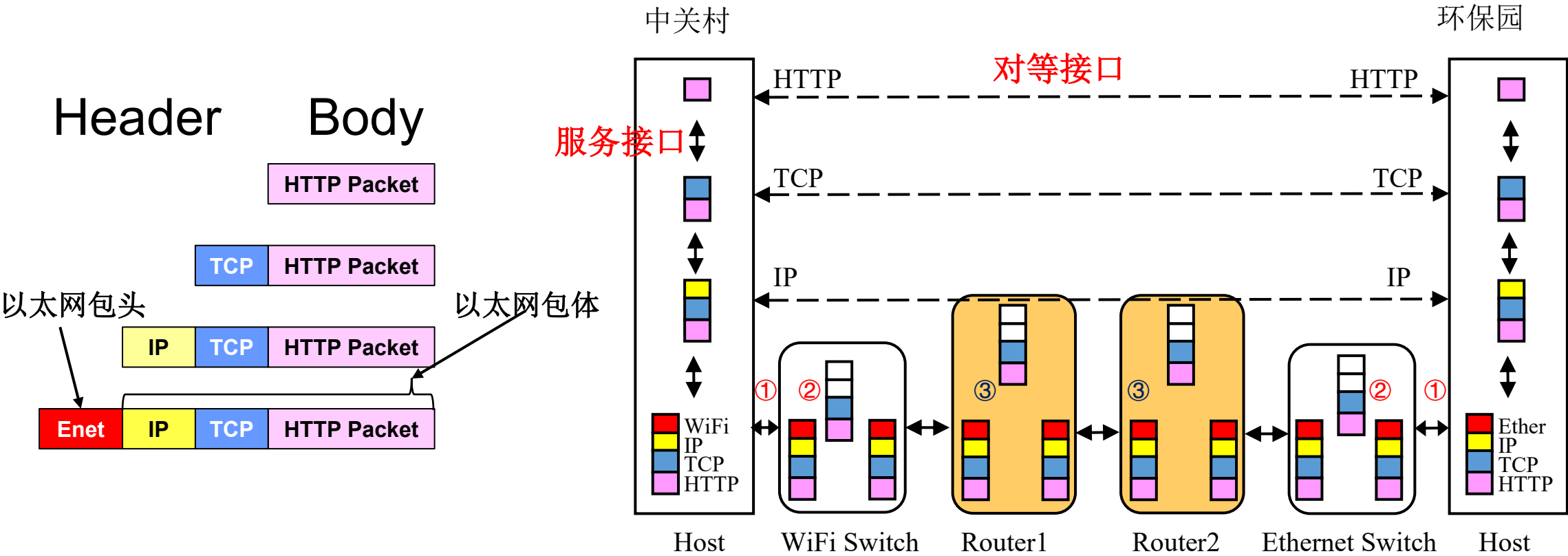
- 从中关村计算所的笔记本电脑通过WiFi访问位于环保园服务器的“猫猫指挥家”图片
- 笔记本电脑IP为192.168.1.101，服务器IP为10.208.104.3，图片URL为：
http://seafire.solid.things.ac.cn:7019/kitty_band/*0555.JPG



协议栈：互联网通信示例



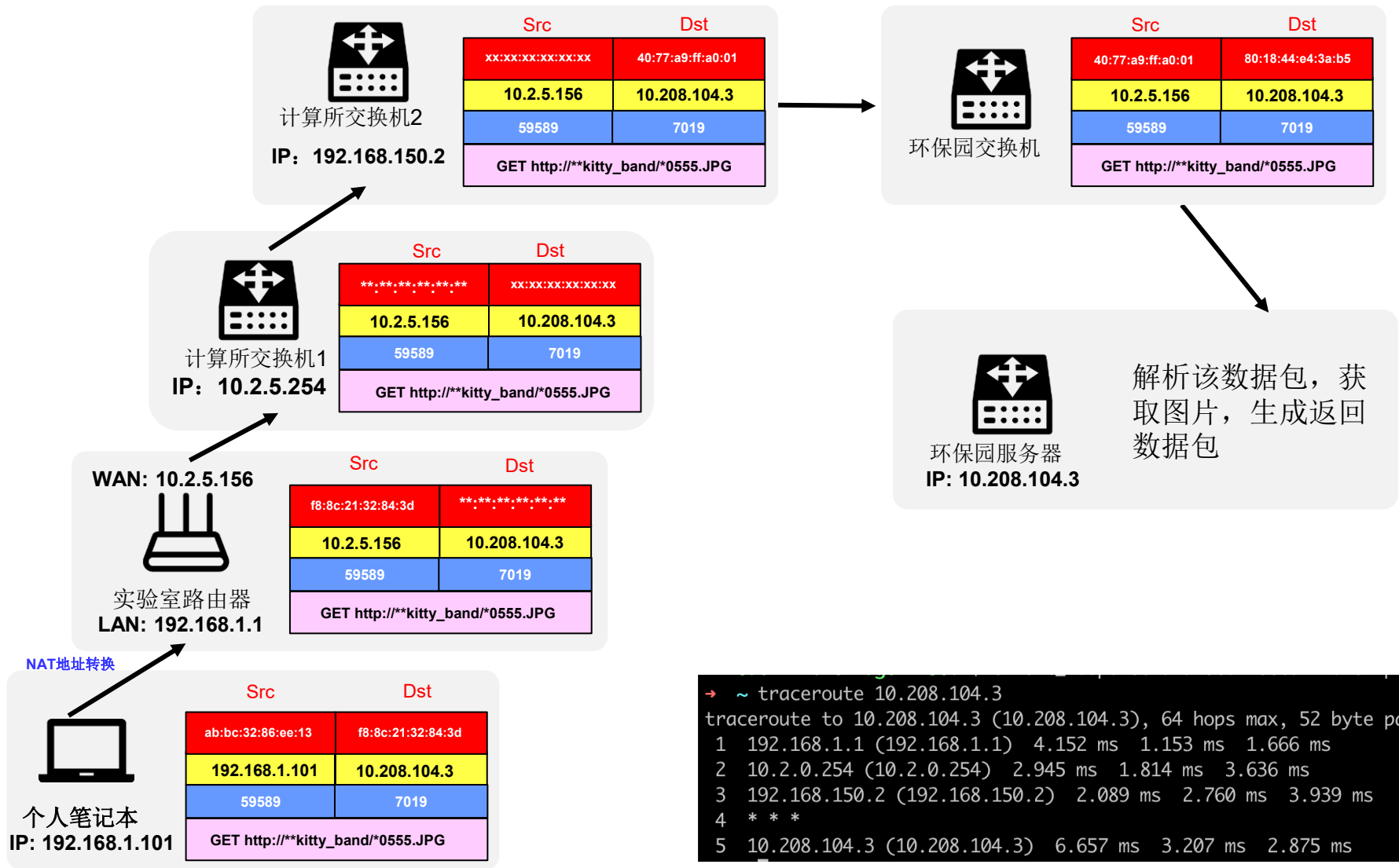
- 从中关村计算所的笔记本电脑通过WiFi访问位于环保园服务器的“猫猫指挥家”图片
- 笔记本电脑IP为192.168.1.101，服务器IP为10.208.104.3，图片URL为：
http://seafile.solid.things.ac.cn:7019/kitty_band/*0555.JPG



From Client to Server



- 个人的笔记本电脑（源IP地址：192.168.1.101）通过WiFi发出请求，经过实验室路由器、计算所交换机和环保园交换机到达环保园服务器（目标IP地址：10.208.104.3）
 - 使用tracert软件追踪从计算所到环保园全路径

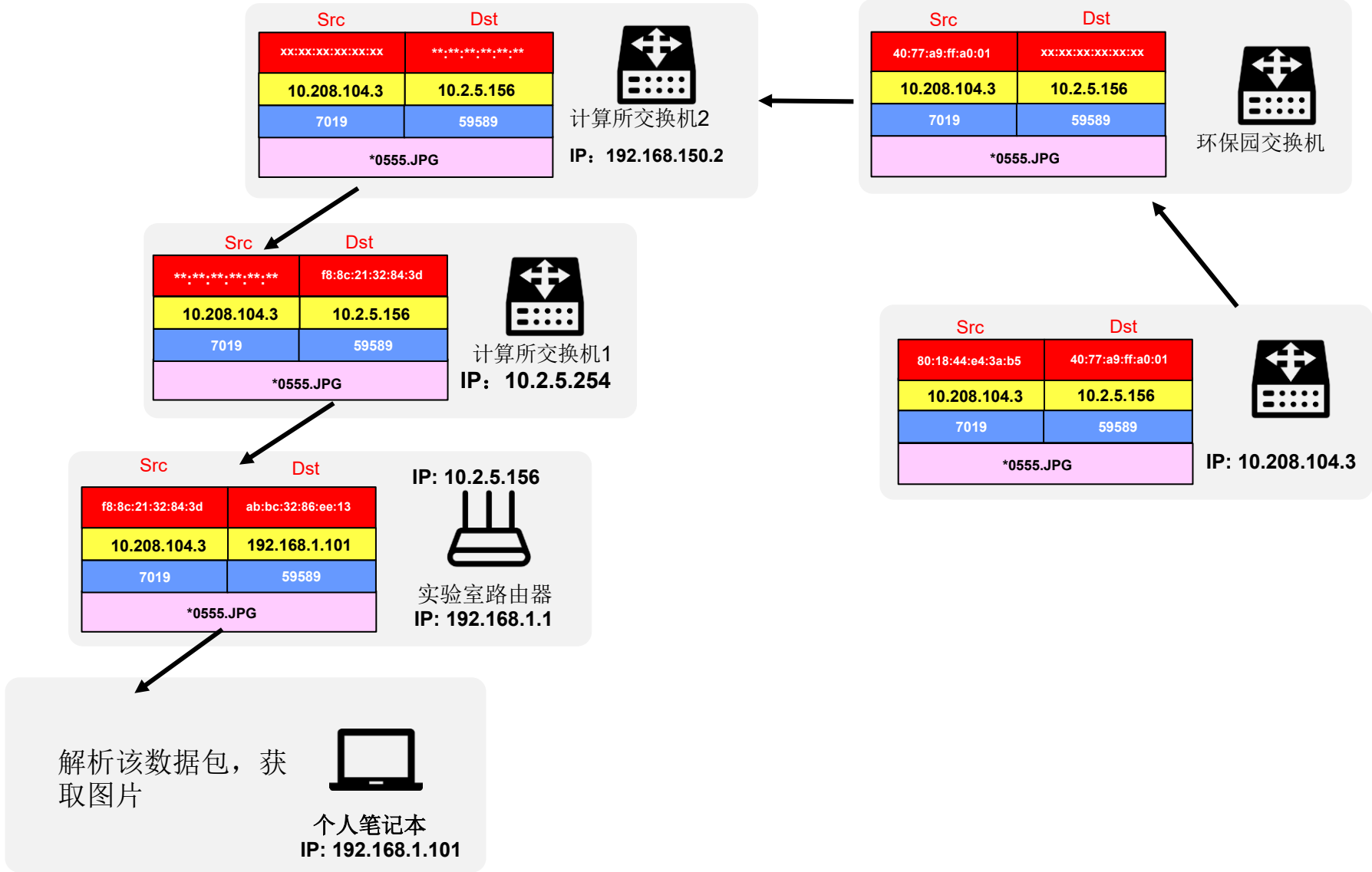


```
→ ~ tracert 10.208.104.3
tracert to 10.208.104.3 (10.208.104.3), 64 hops max, 52 byte packets
 1 192.168.1.1 (192.168.1.1) 4.152 ms 1.153 ms 1.666 ms
 2 10.2.0.254 (10.2.0.254) 2.945 ms 1.814 ms 3.636 ms
 3 192.168.150.2 (192.168.150.2) 2.089 ms 2.760 ms 3.939 ms
 4 * * *
 5 10.208.104.3 (10.208.104.3) 6.657 ms 3.207 ms 2.875 ms
```

From Server to Client



- 环保园的服务器收到请求后，获取图片，生成响应消息的数据包，并返回给个人笔记本
- 使用tracert软件追踪从计算所到环保园全路径

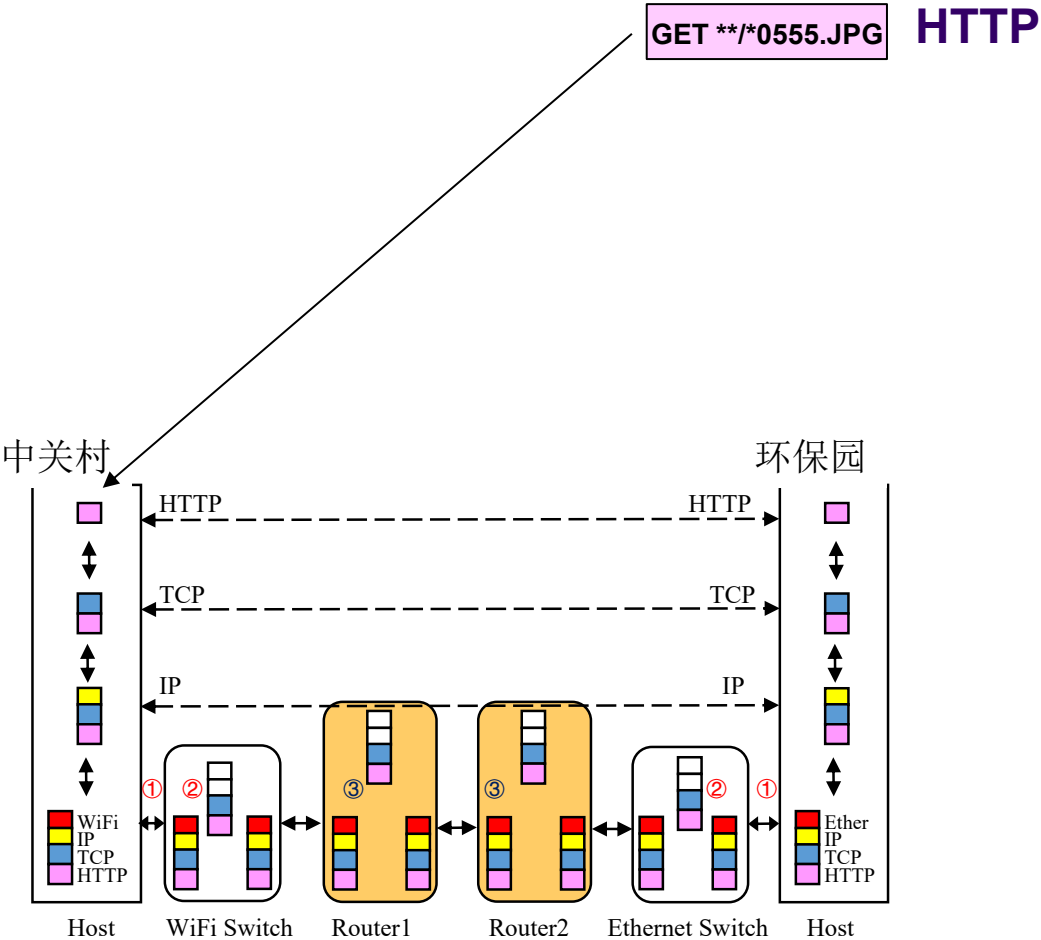


请求消息



- 从中关村计算所的笔记本电脑通过WiFi访问位于环保园服务器的“猫猫指挥家”图片
- 笔记本电脑IP为192.168.1.101 ， 服务器IP为10.208.104.3， 图片URL为：
http://seafire.solid.things.ac.cn:7019/kitty_band/*0555.JPG

请求消息

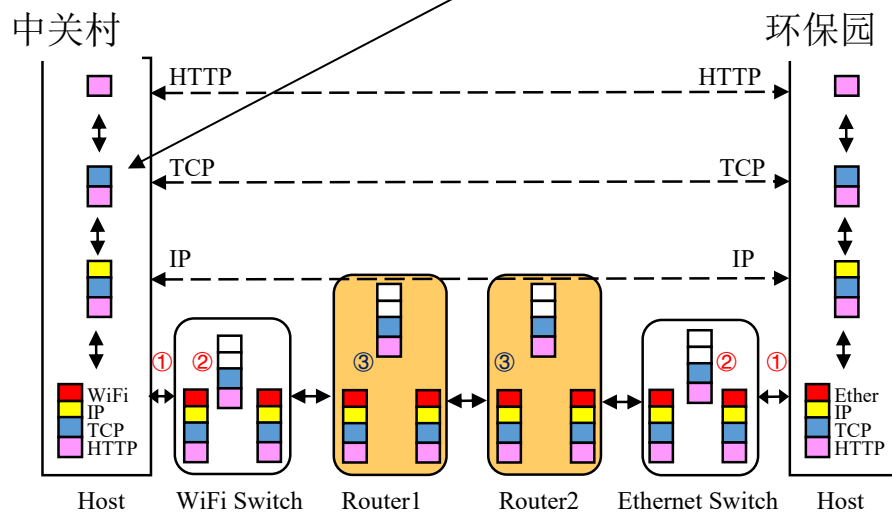
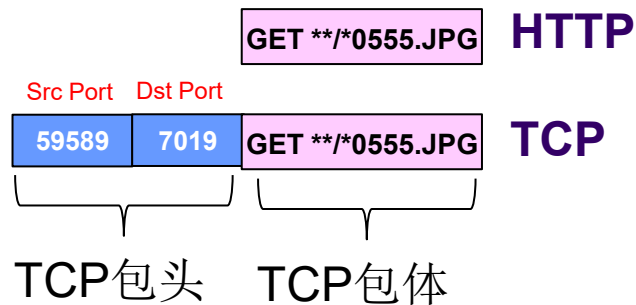


请求消息



- 从中关村计算所的笔记本电脑通过WiFi访问位于环保园服务器的“猫猫指挥家”图片；
- 笔记本电脑IP为192.168.1.101，服务器IP为10.208.104.3，图片地址为：
http://seafle.solid.things.ac.cn:7019/kitty_band/*0555.JPG

- 包头包含三类信息：地址、查错、控制信息
- 本例忽略了TCP包头中的查错信息与控制信息



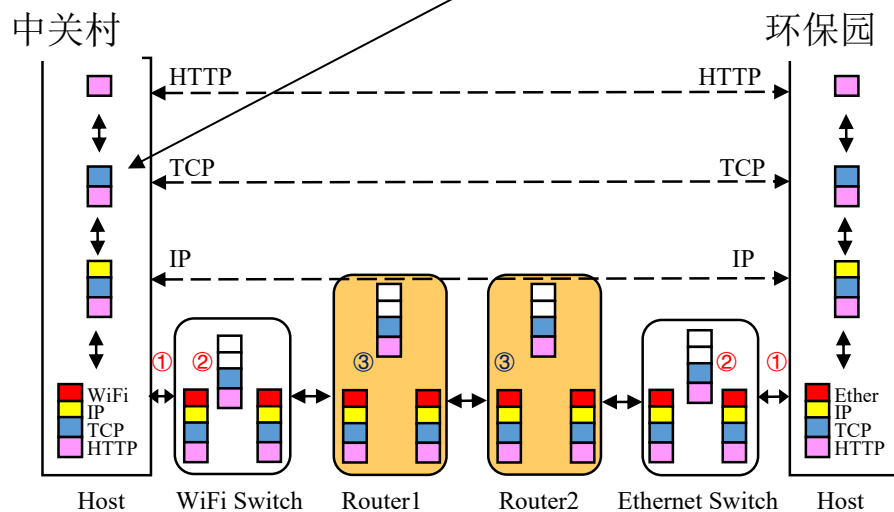
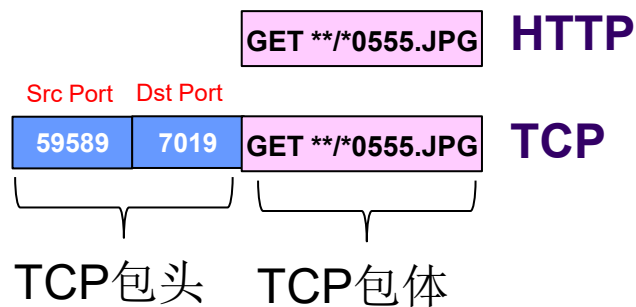
请求消息



- 从中关村计算所的笔记本电脑通过WiFi访问位于环保园服务器的“猫猫指挥家”图片；
- 笔记本电脑IP为192.168.1.101，服务器IP为10.208.104.3，图片地址为：
http://seafile.solid.things.ac.cn:7019/kitty_band/*0555.JPG

- 包头包含三类信息：地址、查错、控制信息
- 本例忽略了TCP包头中的查错信息与控制信息

- Src: Source 源
- Dst: Destination 目的地、目标

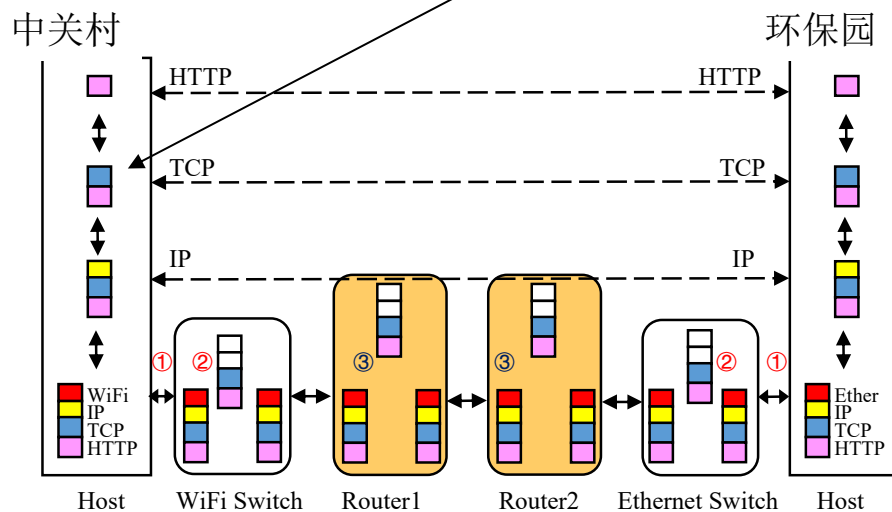
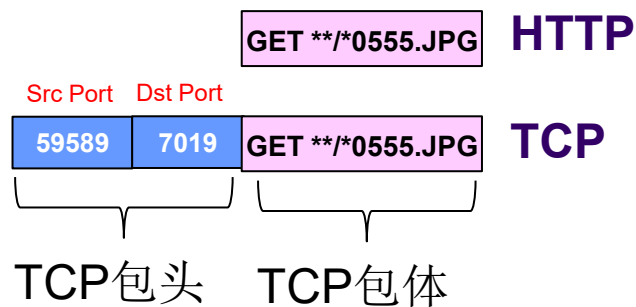


请求消息



- 从中关村计算所的笔记本电脑通过WiFi访问位于环保园服务器的“猫猫指挥家”图片；
- 笔记本电脑IP为192.168.1.101，服务器IP为10.208.104.3，图片地址为：
http://seafile.solid.things.ac.cn:7019/kitty_band/*0555.JPG

- 包头包含三类信息：地址、查错、控制信息
- 本例忽略了TCP包头中的查错信息与控制信息
- 目的地端口7019指代环保园服务器计算机上的Web服务器进程
- 源端口号59589指代中关村笔记本电脑上的浏览器进程



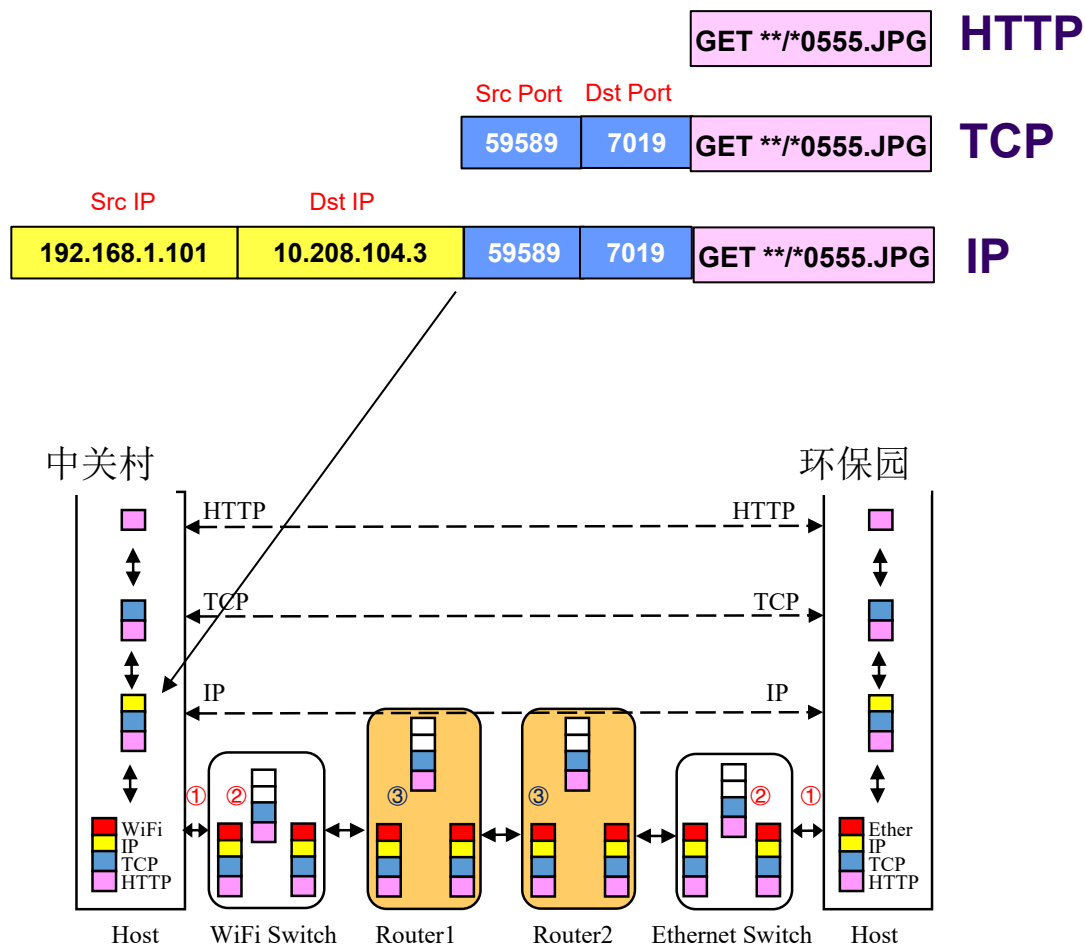
请求消息



- 从中关村计算所的笔记本电脑通过WiFi访问位于环保园服务器的“猫猫指挥家”图片；
- 笔记本电脑IP为192.168.1.101，服务器计算机IP为10.208.104.3，图片地址为：
http://seafile.solid.things.ac.cn:7019/kitty_band/*0555.JPG

- 10.208.104.3是内网IP地址 (***)

- 哪一段是IP包体？
- 哪一段是IP包头？



请求消息

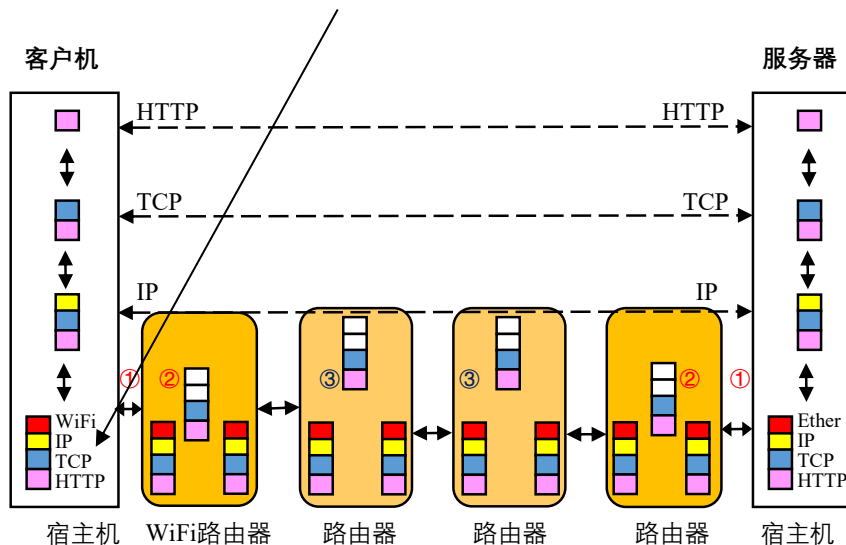
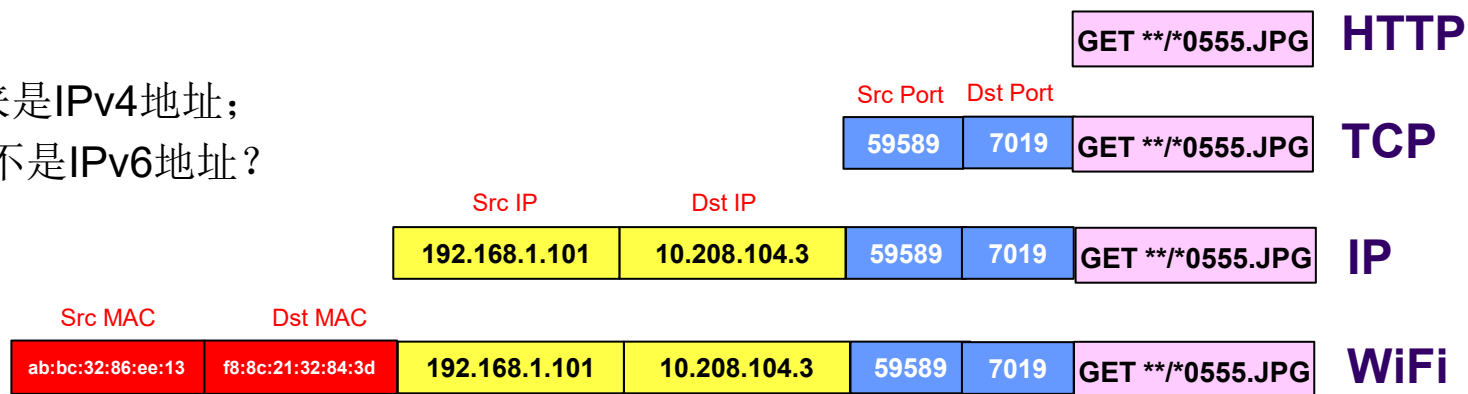


- 从中关村计算所的笔记本电脑通过WiFi访问位于环保园服务器的“猫猫指挥家”图片；
- 笔记本电脑IP为192.168.1.101，服务器IP为10.208.104.3，图片地址为：
http://seafile.solid.things.ac.cn:7019/kitty_band/*0555.JPG

- 数据链路层：那一段是包头，那一段是包体？

192.168.1.101看起来是IPv4地址；

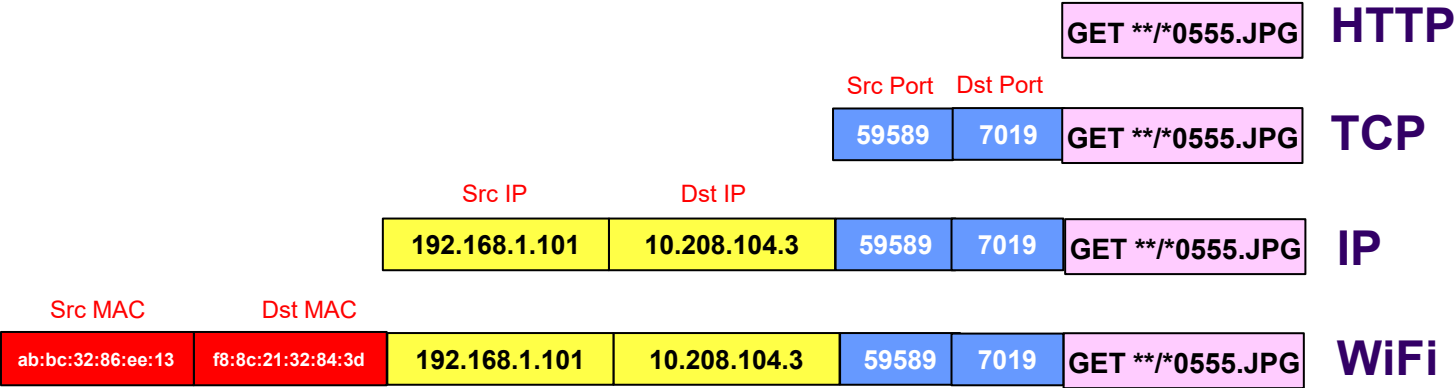
f8:8c:21:32:84:3d是不是IPv6地址？



请求消息：四层的包格式与内容

- 从中关村计算所的笔记本电脑通过WiFi访问位于环保园服务器的“猫猫指挥家”图片；
- 笔记本电脑IP为192.168.1.101，服务器IP为10.208.104.3，图片地址为：
http://seafile.solid.things.ac.cn:7019/kitty_band/*0555.JPG

该层的消息包
确定了哪些信息？



应用载荷

+ 进程

+ 互联网主机

+ 局域网主机

