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

计算机科学导论

网络思维

三个课时

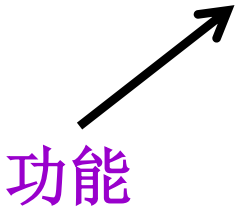
徐志伟
中科院计算所
zxu@ict.ac.cn

提纲

1. 计算过程与网络思维
    1. 连通性、消息传递协议
2. 网络思维例子
3. 网络思维要点
    1. 名字空间
    2. 拓扑：网络是图（节点+连接）
    3. 协议栈
4. 服务质量（即使出现错误或故障）
5. 网络的规律

网络思维与算法思维有很大的不同

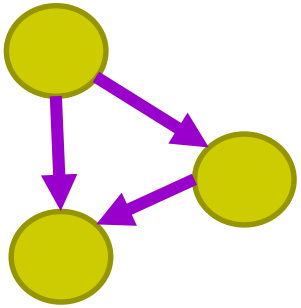
- 算法有统一的定义，网络有很多定义
  - 算法的定义【**Donald Knuth**】. 一个算法是一组有穷的规则，给出**求解特定类型问题**的运算序列，并具备下列五个特征：
    - **有限性**：一个算法在有限步骤之后必然要终止。
    - **确定性**：一个算法的每个步骤都必须精确地（严格地和无歧义地）定义。
    - **输入**：一个算法有零个或多个输入。
    - **输出**：一个算法有一个或多个输出。
    - **能行性**：一个算法的所有运算必须是充分基本的，原则上人们用笔和纸可以在有限时间内精确地完成它们。



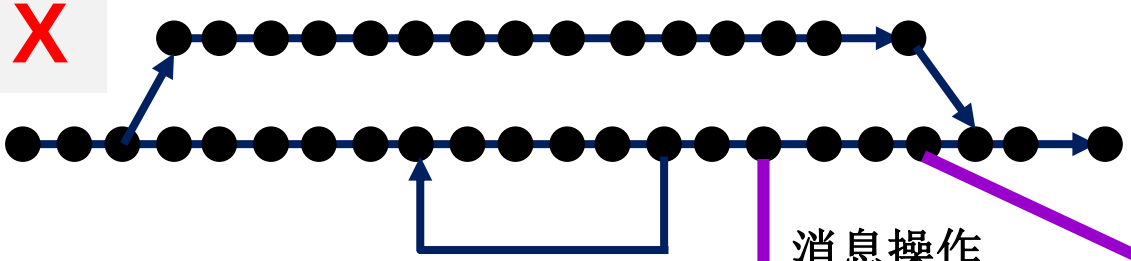
- 算法有时间复杂度等少数几个统一指标；网络有很多指标
- 算法由基本运算与步骤组合而成，时间复杂度具有组合性；网络的指标组合性较弱

1. 网络思维的主要知识点

- 计算过程涉及（由多个**节点**连接而成的）**网络**
  - 网络成为计算过程的对象、执行系统
- 核心概念：连通性、消息传递、协议
  - **名字空间、拓扑、协议栈**



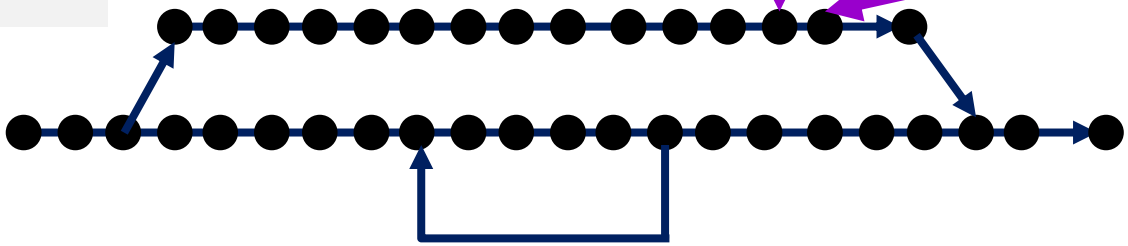
X



消息操作

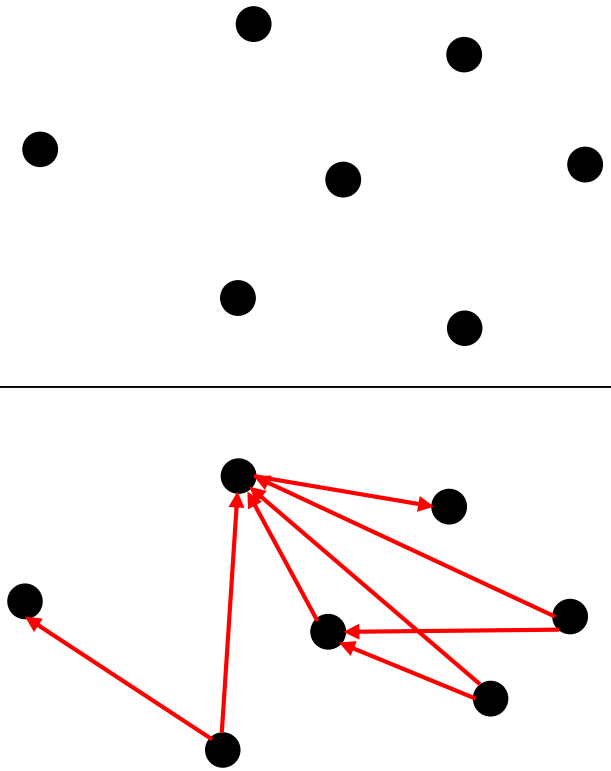
Z

Y



连通性与消息传递是松耦合关系

- 网络思维并不必涉及消息传递（或通信协议）
- 此时，重点是**连通性**（connectivity）
  - 即：有什么节点？节点之间如何连接？
  - 拓扑本身就有价值
- 搜索引擎实例
  - 第一代：无网络思维
    - 只关心节点的内容
  - 第二代：有网络思维
    - Page、 Kleinberg、李彦宏
    - 关心节点内容
    - **还关心网络拓扑（pagerank）**

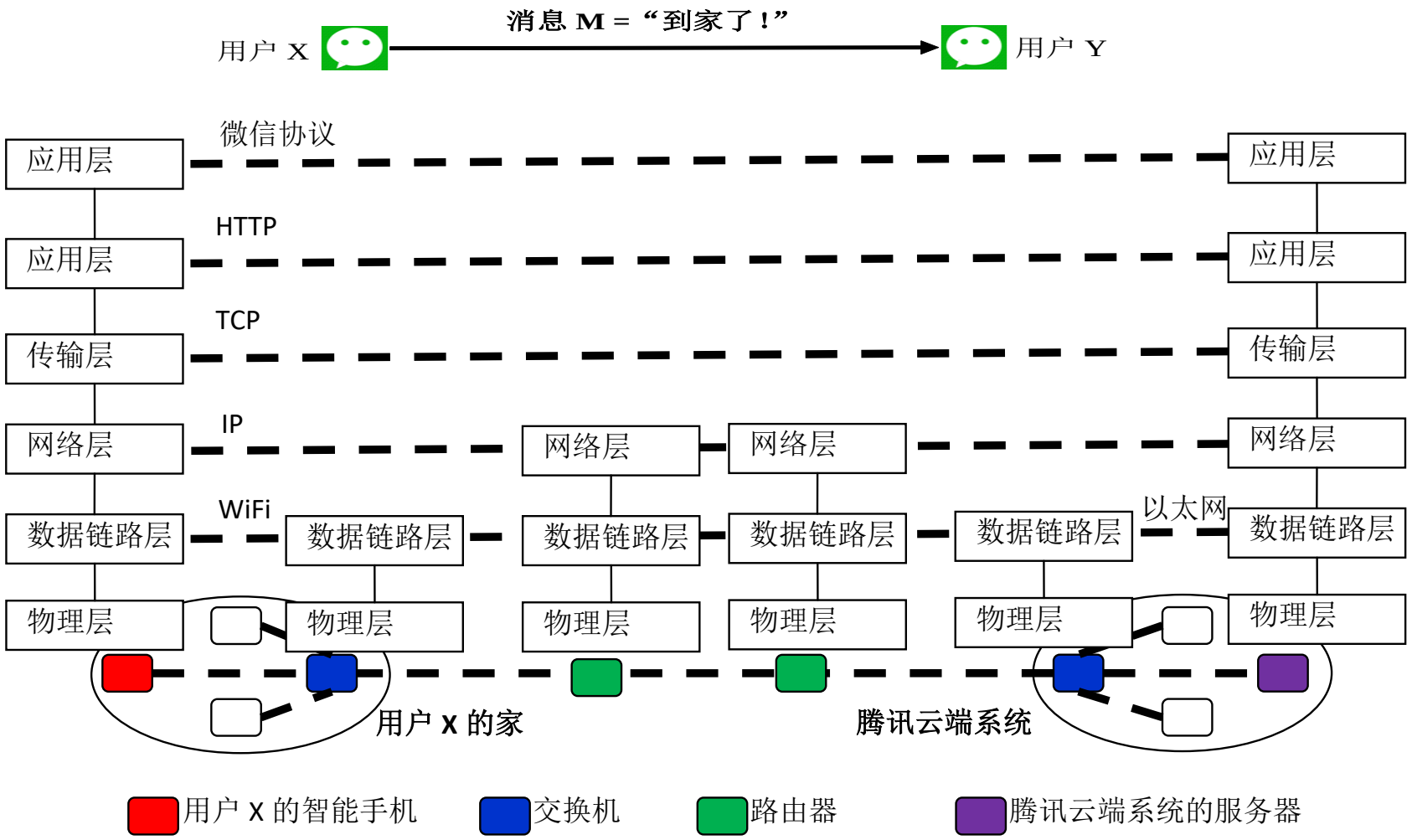


用网络思维解决问题的另一个实例

- 传统的电话通信系统
  - 采用专用的通信信令协议栈
    - 保证语音通信质量
    - 成本高
- 今天很多通信系统采用通用互联网协议栈实现
  - 挑战：保证质量
  - 优点：利用全球互联网
  - 构建在互联网之上 → Over The Top（简称OTT）
  - “OTT服务”是网络思维实例，成功例子很多
    - 微信
    - 视频
    - 语音

2. 从微信例子看协议

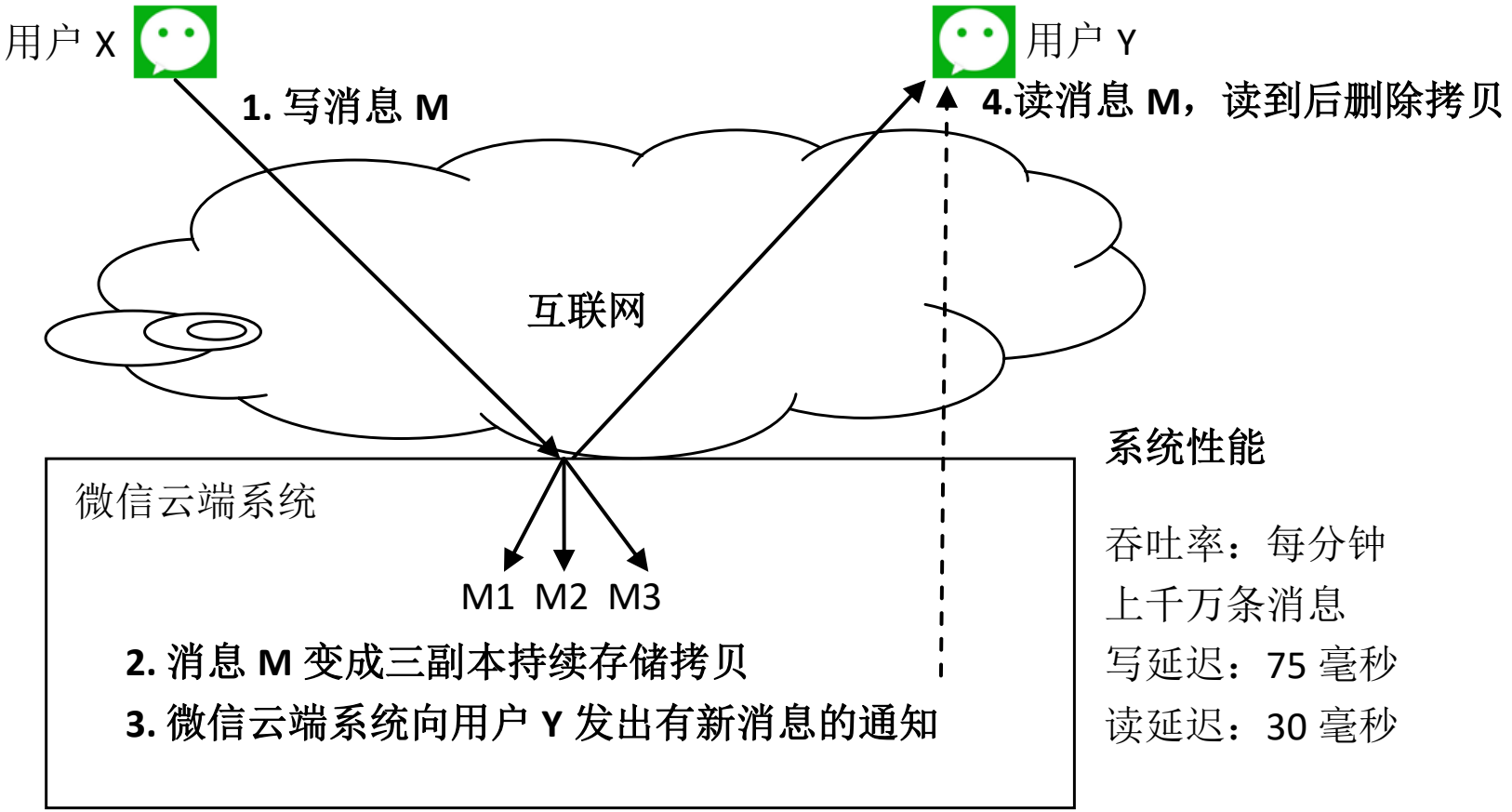
- 网络：多个（算法+通信操作）组合而成



微信网络协议栈（实线表示层间接口，虚线表示对等接口）

微信系统的实际情况示意

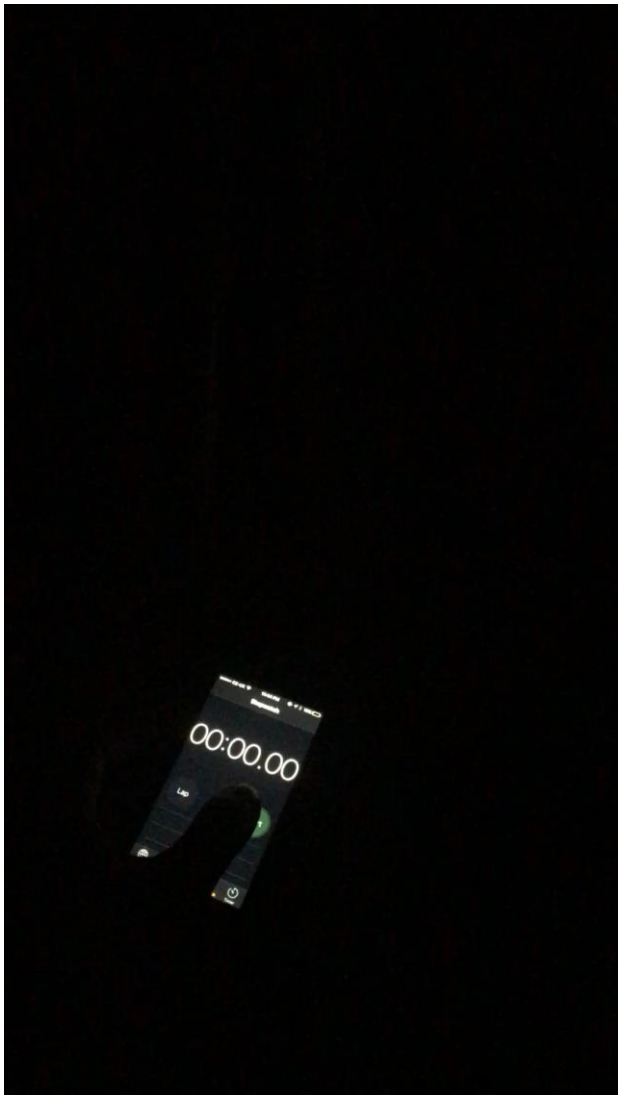
- 网络思维有助于回答：“为什么不是简单的用户直接通信，而是更为复杂的实际系统？”



3. 网络思维要点

- 网络思维：是连通性和协议的整体
  - 是名字空间、网络拓扑、协议栈的整体
- 三个核心概念
  - 名字空间：精确的说明一个网络有哪些节点
    - 通信“对方”是谁、连接“伙伴”是谁
  - 拓扑：哪些节点间需要连接和通信？
  - 协议栈：节点间如何连接与通信，甚至做更高级的操作？
    - 传递消息：互联网
    - 传递信任？区块链
    - 传递情感？社交网络部分提供
    - 传递智能？

Amazon Echo设备控制实测



- **测试环境：**Amazon Echo Dot, 英国50Mbps宽带，Yeelight米家智能LED台灯。
- **总用时：**用户语音完毕->Echo响应OK：3.96s
  - 用户语音完毕->Yeelight平台向台灯发出指令(语音识别与分析): 约3.7s
  - 台灯执行关灯->返回给Yeelight平台(指令下发):0.16s
  - Yeelight平台收到确认到Echo返回OK:0.1s
- **传输路径：**伦敦 -> 美国亚马逊Alexa-> Yeelight新加坡AWS -> 伦敦 -> Yeelight新加坡AWS -> 美国亚马逊Alexa-> 伦敦

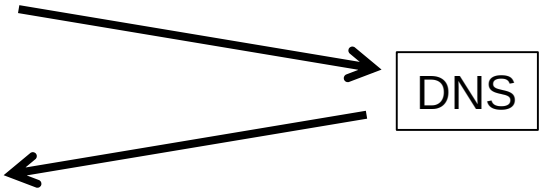
名字空间

- Name space；naming
- 主要用于指称网络中的节点

名字空间实例	节点的名字举例	名字空间解释
微信名字	中关村民	腾讯公司规定的任意“合法的”字符串
电子邮箱地址	zxu@ict.ac.cn	用户名@因特网域名
手机号码	189-8888-9999	通信公司规定的11位10进制数字串
本机文件路径（本地路径）	/我的文件/教材.pdf	本机操作系统规定的文件名
本机网卡地址（MAC地址）	00-1E-C9-43-24-42	全球统一规定的12位16进制数字串
网站域名	www.ict.ac.cn	互联网协议栈规定的域名
网站IP地址	159.226.97.84	IP协议规定的合法地址

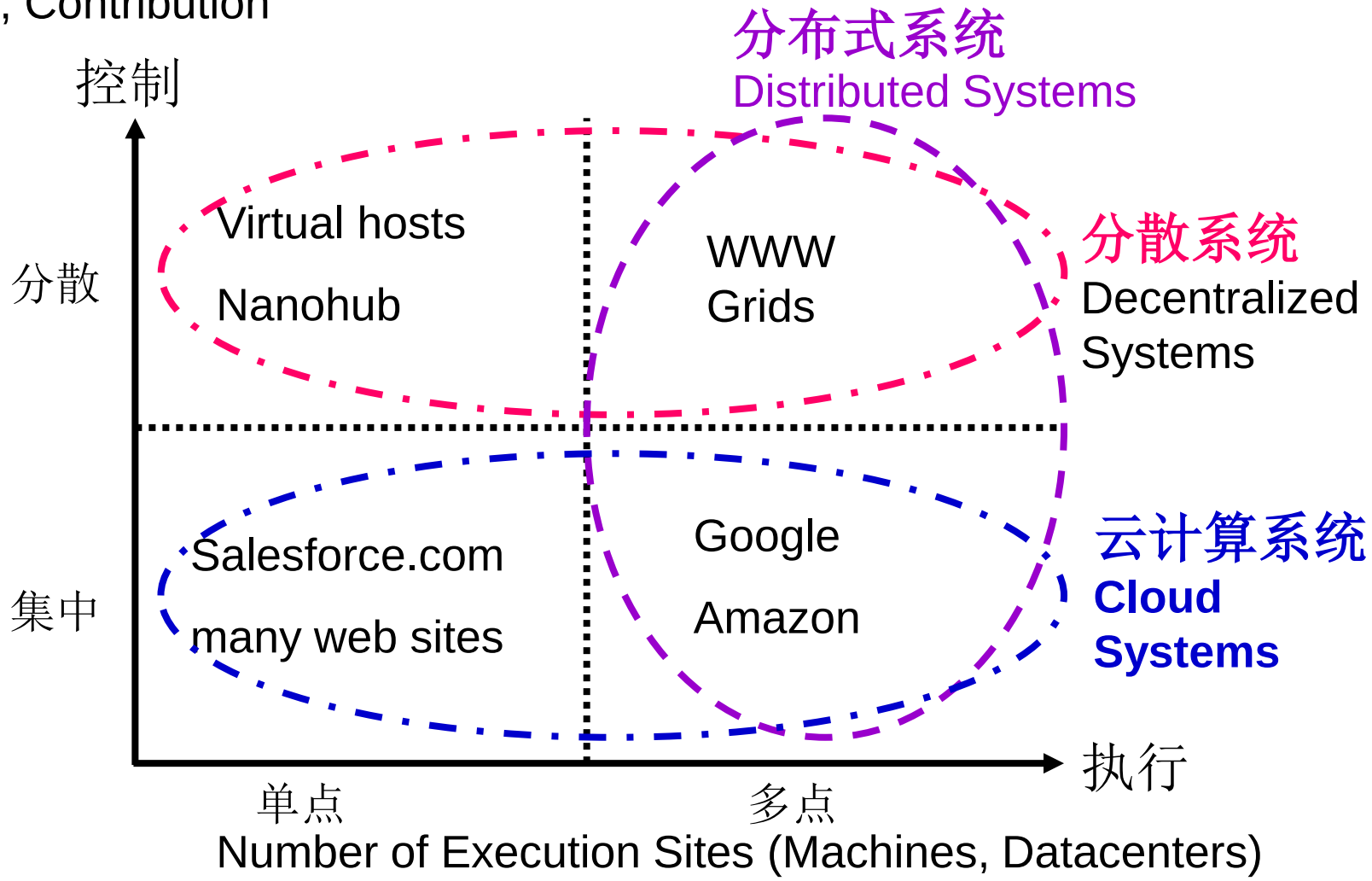
名字空间设计的两个问题

- 设计与理解名字空间的基本考虑
  - **唯一性**：zxu@ict.ac.cn　vs.　中关村民
    - 重用性：手机号码 vs. 万维网资源的URI
  - **动态性**：一个图形加速卡可以插到另一台设备总线上吗？
    - 固定IP地址 vs. 动态生成IP地址
  - **友好性**：中关村民 vs. 以太网MAC地址
- 不同层次间的名字如何解析
  - 本地解析 vs. 全网解析（远程解析）
    - http://www.ict.ac.cn/本地路径...
    - http://159.226.97.84/本地路径...



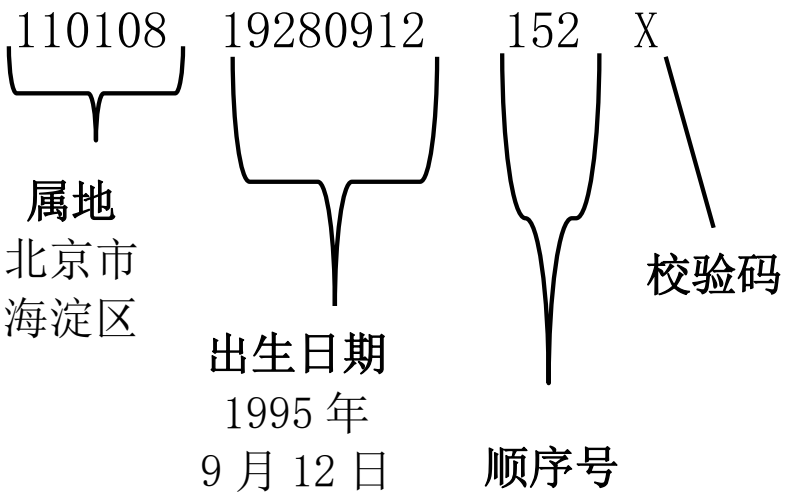
命名是基本功能——网络计算系统分类

Admin, Knowledge, Naming,
Coding, Contribution



第二代身份证18位数字名字空间

- 名字空间的性质
  - 能否保证名字唯一性？
  - 名字的自主性如何？
  - 友好性如何？
  - 11010819560921141的解释和验证码



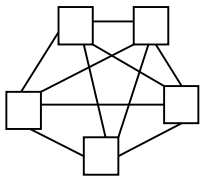
余数	校验码
0	1
1	0
2	X
3	9
4	8
5	7
6	6
7	5
8	4
9	3
10	2

网络拓扑

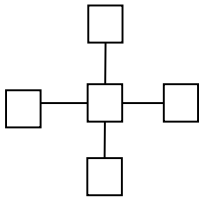
- 网络往往可以看成是一个由节点和连线（也称为边）构成的图
  - 网络不是节点的集合
  - 搜索引擎发展实例（数十亿用户、千亿美元市场）
    - Alta Vista vs. Google（Page）、Kleinberg、百度（李彦宏）
- 设计与理解网络拓扑的基本考虑
  - 可扩展性
    - 规模可扩展（10个、10亿个节点）
    - 地域可扩展（局域网、广域网等）
  - 连通性与容错
  - 节点的连接度、节点间距离、网络的直径
    - 消息通信的延迟、带宽、成本、功耗
  - 动态性：静态网络、动态网络、演化网络

按动态性划分的三类网络拓扑

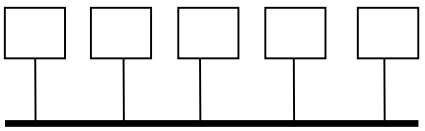
- 静态网络：节点完全确定、连接完全确定
- 动态网络：节点完全确定、连接部分确定
- 演化网络：节点部分确定、连接部分确定
- **你的微信朋友圈是什么网络？**



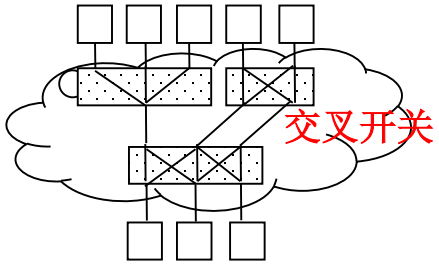
(a) 全连通图



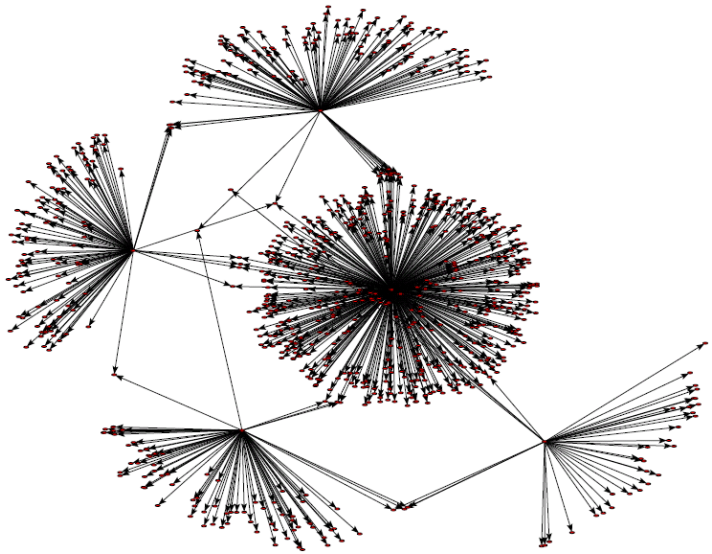
(b) 星型网络



(c) 总线



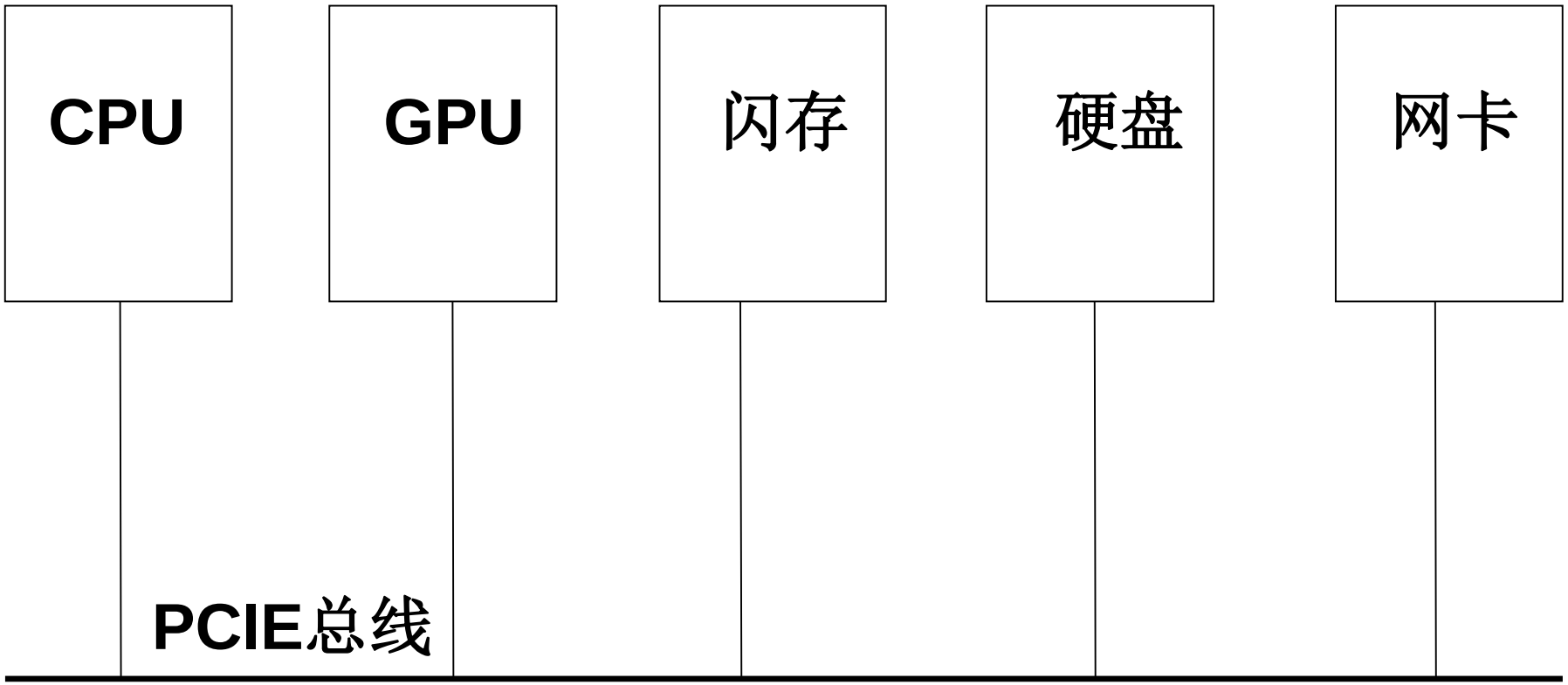
(d) 交换网络



(e) 演化网络

名字空间、拓扑、协议栈

- 三者如何合作实现互联通信。三者是一体。
- 总线仲裁实例



对计算思维的理解之六：连通性

- 连接体通常被称为**网络**。网络不只是连线

计算过程与计算思维理解之六：连通性

很多问题涉及用户/数据/算法的连接体，而非单体

计算过程刻画：

- 一个计算过程是解决某个问题的有限个计算步骤的执行序列。
- 计算过程的整体或一个步骤可能需要将连接体作为处理对象。
- 计算过程的整体或一个步骤可能需要在连接体上执行。

计算思维要点：“精准地描述信息变换过程的操作序列，并**有效地**解决问题”是如何体现的？

- 用名字空间和网络拓扑精准地描述连接体，即操作对象或执行系统。
- 在问题建模或解题过程中，不只使用单点做计算，而是采用连接体（即多个节点互联而成的网络）作为计算对象或计算系统。

分布式算法实践：实例1

- 问题：求出全班男同学数，女同学数，同学总数
  - 要点：独立求出这三个总数（X, Y, Z），以便容错
- 算法1
  - 全体同学站起来；每个同学设定初始的（X, Y, Z）
    - 男同学设定(X, Y, Z) := (0, 1, 1); 女同学设定(X, Y, Z) := (1, 0, 1);
  - 仍然站立的每两个同学（A, B）将各自的三元组加起来
    - A同学随意找一个站立的同学作为B同学， B同学接受；
    - A同学将自己的三元组传给B同学；
    - B同学收到后与自己的三元组相加，形成新的三元组
      - 注意检查相加结果是否正确，如不正确要求A同学重传
    - A同学坐下
  - 最后唯一站立的同学报出三元组之和

分布式算法实践：实例2

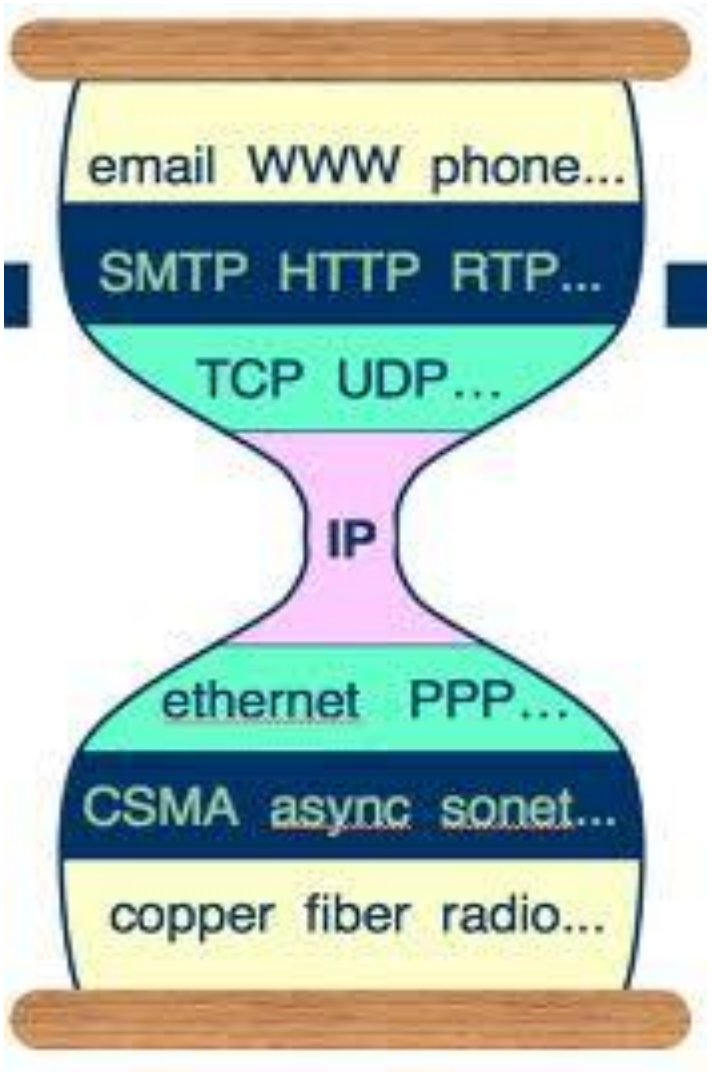
- 问题：求出全班男同学数，女同学数，同学总数
  - 要点：独立求出这三个总数（X, Y, Z），以便容错
- 算法2
  - 与算法1基本相同，但只用局部通信，即：
    - 每一排的同学先求出该排三元组之和，放在第一列；
    - 然后再求出第一列三元组之和，放在第一排第一列同学

- 算法1、算法2分别用了多少步？
- 假如全班有400万名同学，算法1、算法2分别会用多少步？

协议栈的生态系统思想
互联网协议的沙漏模型

- IP是细腰
- 2000年以来，
  HTTP也是细腰
- OTT (over the top)服务

- 支持多种上层应用
- 支持多种下层技术



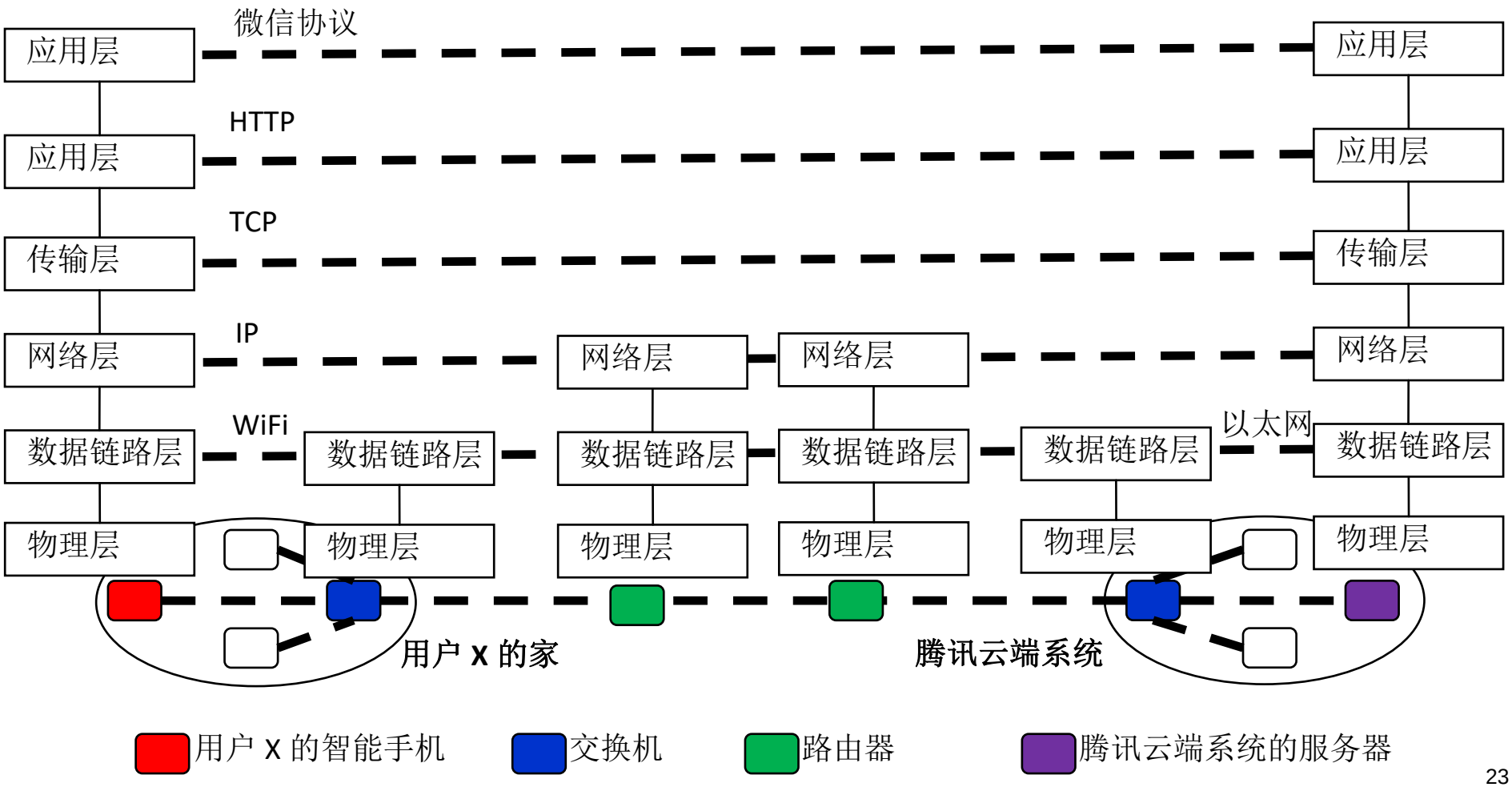
图片来源： Famous Internet Quotes，
https://www.ofcourseimright.com/?page_id=1504

从应用角度理解协议栈的四个问题

- 通信过程涉及互联网协议栈的哪些接口？
- 通信过程涉及互联网协议栈的哪些层次？
- 通信过程涉及哪些硬件？
- 一条应用层的微信消息“到家了！”如何解析成底层的消息包？

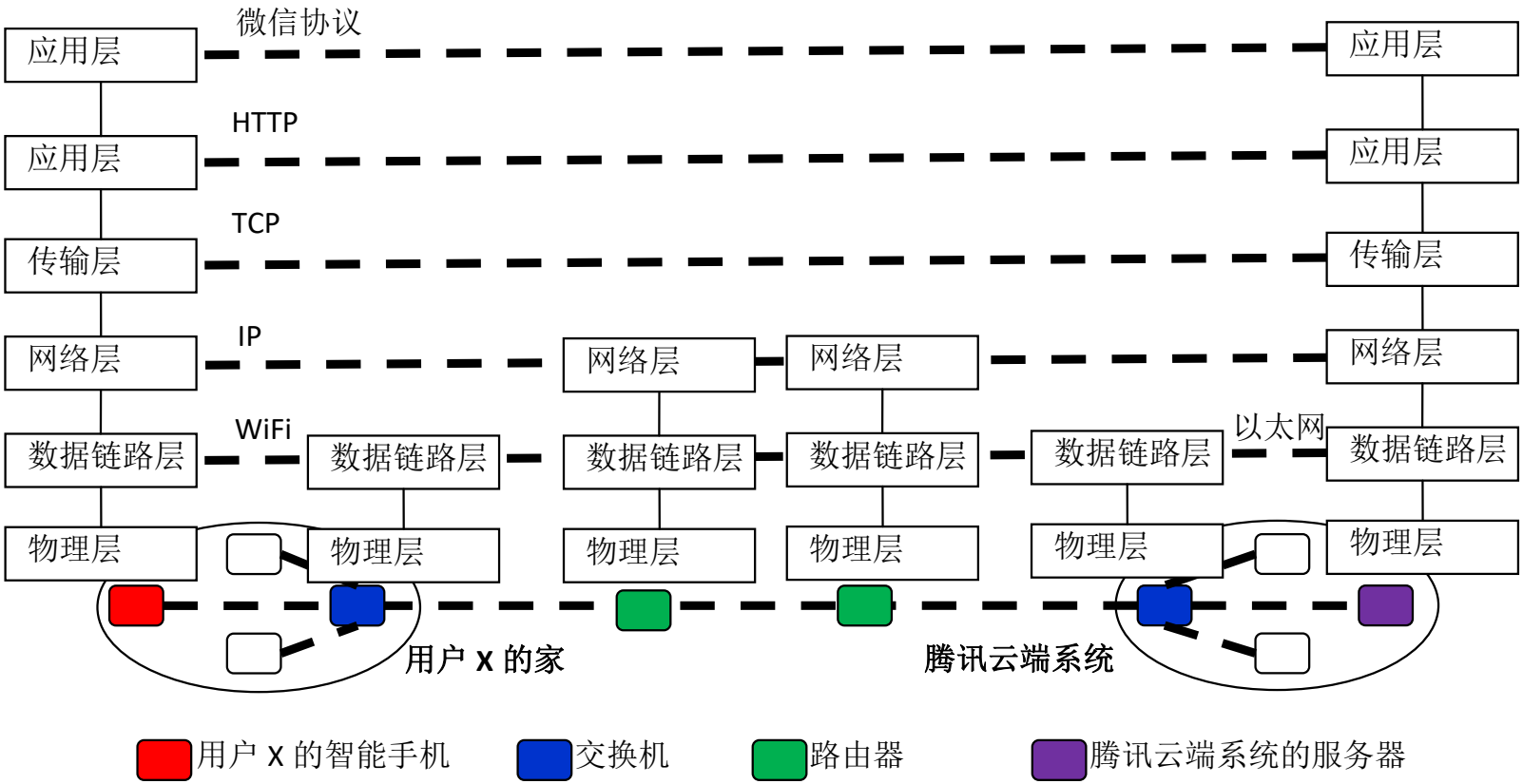
通信过程涉及互联网协议栈的哪些接口？

- 对等接口、层间接口



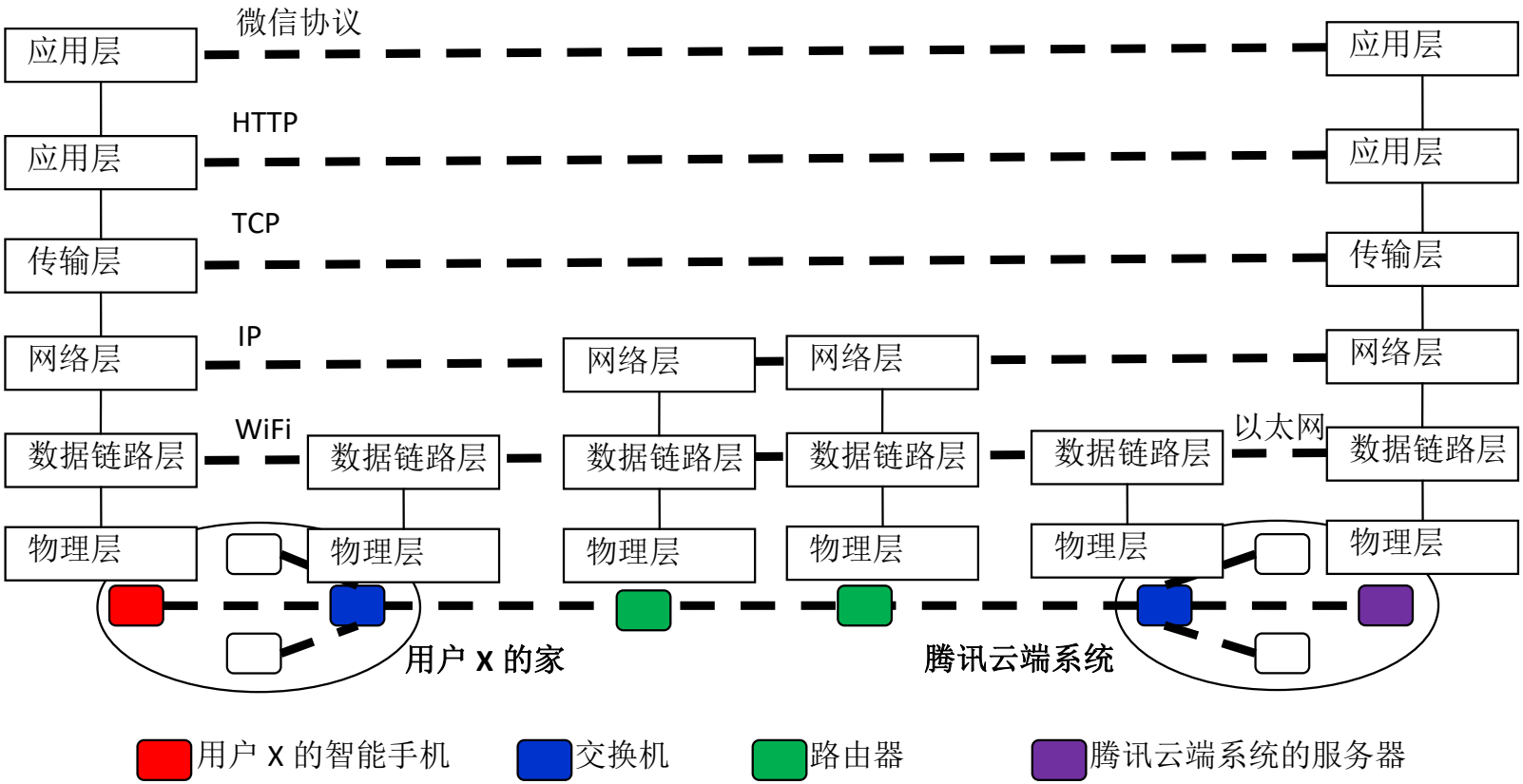
通信过程涉及互联网协议栈的哪些层次？

- 应用层：　　到应用
- 传输层：　　到进程
- 网络层：　　到跨网设备
- 数据链路层、物理层：　　到网内设备



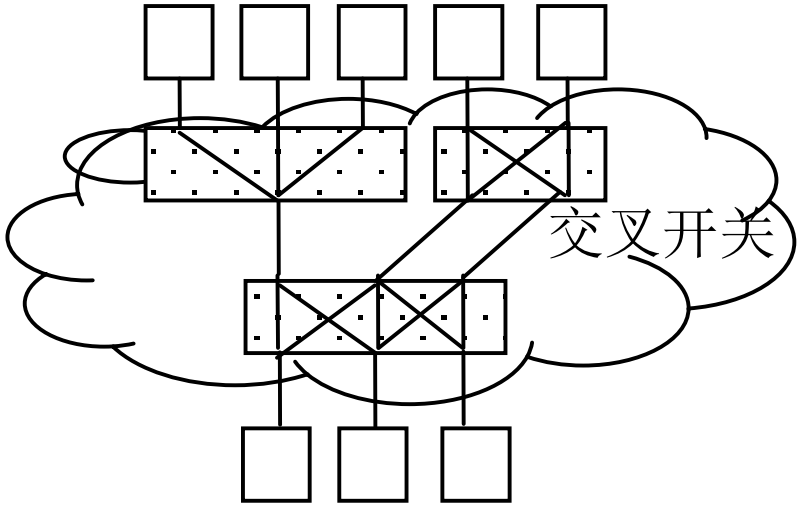
通信过程涉及哪些硬件？

- 用户的智能手机、家庭无线网（WiFi）、WiFi交换机、广域网、两个IP路由器、腾讯云端系统交换机、腾讯云端系统数据中心局域网、腾讯云端系统服务器
- 为什么WiFi交换机通常被称为WiFi路由器？



“到家了！”如何解析成底层的消息包？

- 关键技术点
  - 线路虚拟化：分组交换
  - 包：首部+数据
  - 逐层解析并传输
- **规范实践与黑客实践**



微信协议：中关村民送消息“回家了！”给好友 Y

应用层		
应用层	HTTP	GET http://mp.weixin.qq.com/...回家了！...
传输层	TCP	TCP 首部 [**80 端口**] | **GET :// /...回家...**
网络层	IP	IP 首部 [**140.206.160.199**] | TCP 首部 | **GET :// /...回家...**
WiFi 层	802.11	WiFi 首部 | IP 首部 | TCP 首部 | **GET :// /...回家...**

对计算思维的理解之七：协议栈

计算过程与计算思维理解之七：协议栈

节点之间通过协议栈传递消息

计算过程刻画：

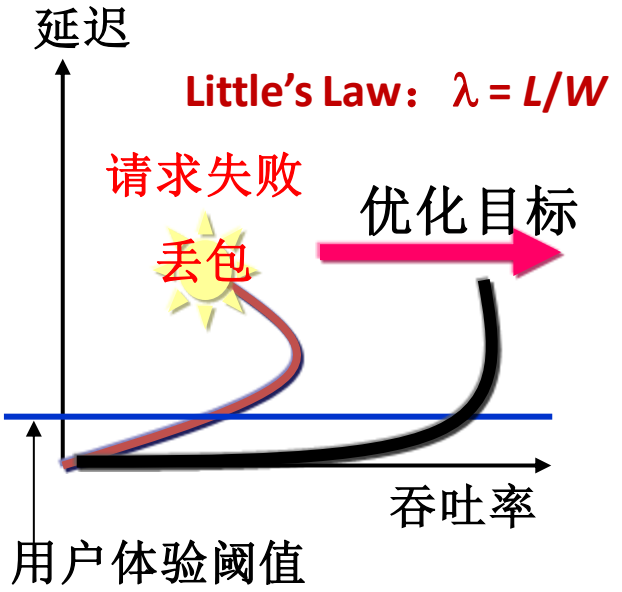
- 计算过程可包含消息传递步骤。
- 消息传递的核心操作是包交换，即通过“包头+包体”的消息包传递信息。

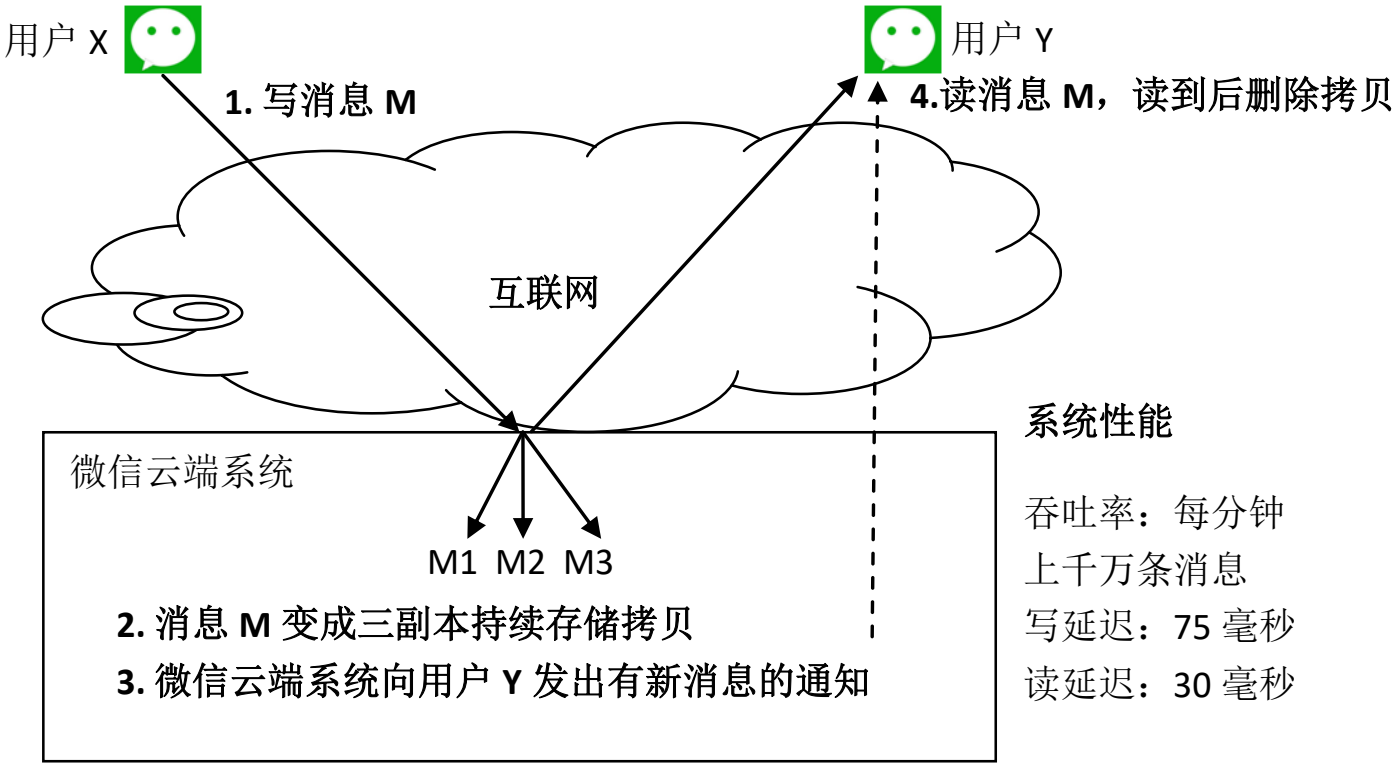
计算思维要点：“精准地描述信息变换过程的操作序列，并**有效地**解决问题”是如何体现的？

- 精准地描述协议栈整体，以及每个协议的消息格式、层间接口、对等接口。
- 针对问题，确定应用协议层次，充分重用通用的互联网协议。

4. 服务质量与用户体验

- 传输单个消息的性能
- 整个网络计算系统的性能
  - 请求规模、延迟、吞吐率





系统性能

吞吐率：每分钟上千万条消息

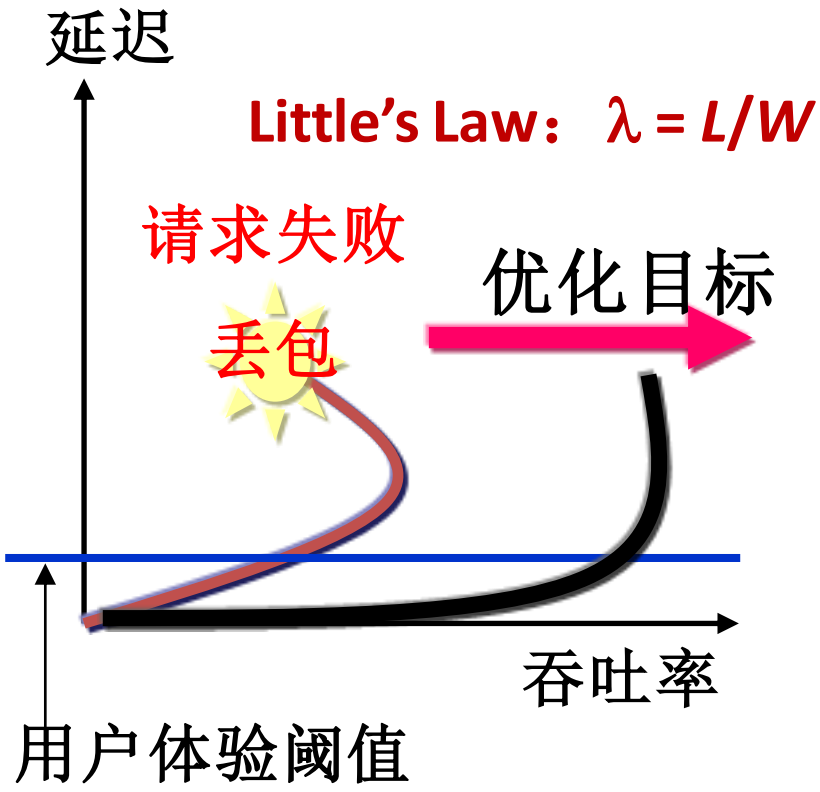
写延迟：75 毫秒

读延迟：30 毫秒

2013.11.11
“光棍节”
微信系统
实现的性能

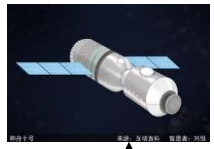
服务质量的多面观

- 服务质量=用户体验
- 用户体验差的功能，是缺失的功能
- 体验差的例子
  - 延迟太长
  - 马赛克
  - 视频卡断
  - 陌生人混入朋友圈
  - 机器翻译质量太差



太空通信的三个问题

- 2013年6月20日王亚平老师太空授课
  - 秒表测试来回延时时间 ≈ 5秒
    - 视频上王亚平老师反应时刻 - 视频上同学语音问题结束时刻
    - 680/30万=2.267毫秒，比5秒小了2000多倍。为什么？
- 历史推演
  - 神舟五号实现了音频对话
  - 神舟九号实现了双向视频可视通话
  - 神舟十号实现了视频太空授课
  - 未来可能实现什么？
    - 延时小于275毫秒的实时视频太空授课？
- 中学生向神舟X号宇航员送礼物
  - 同学们制作的神舟飞船小模型？
  - 送一朵鲜花？



340
公里

网络的规律

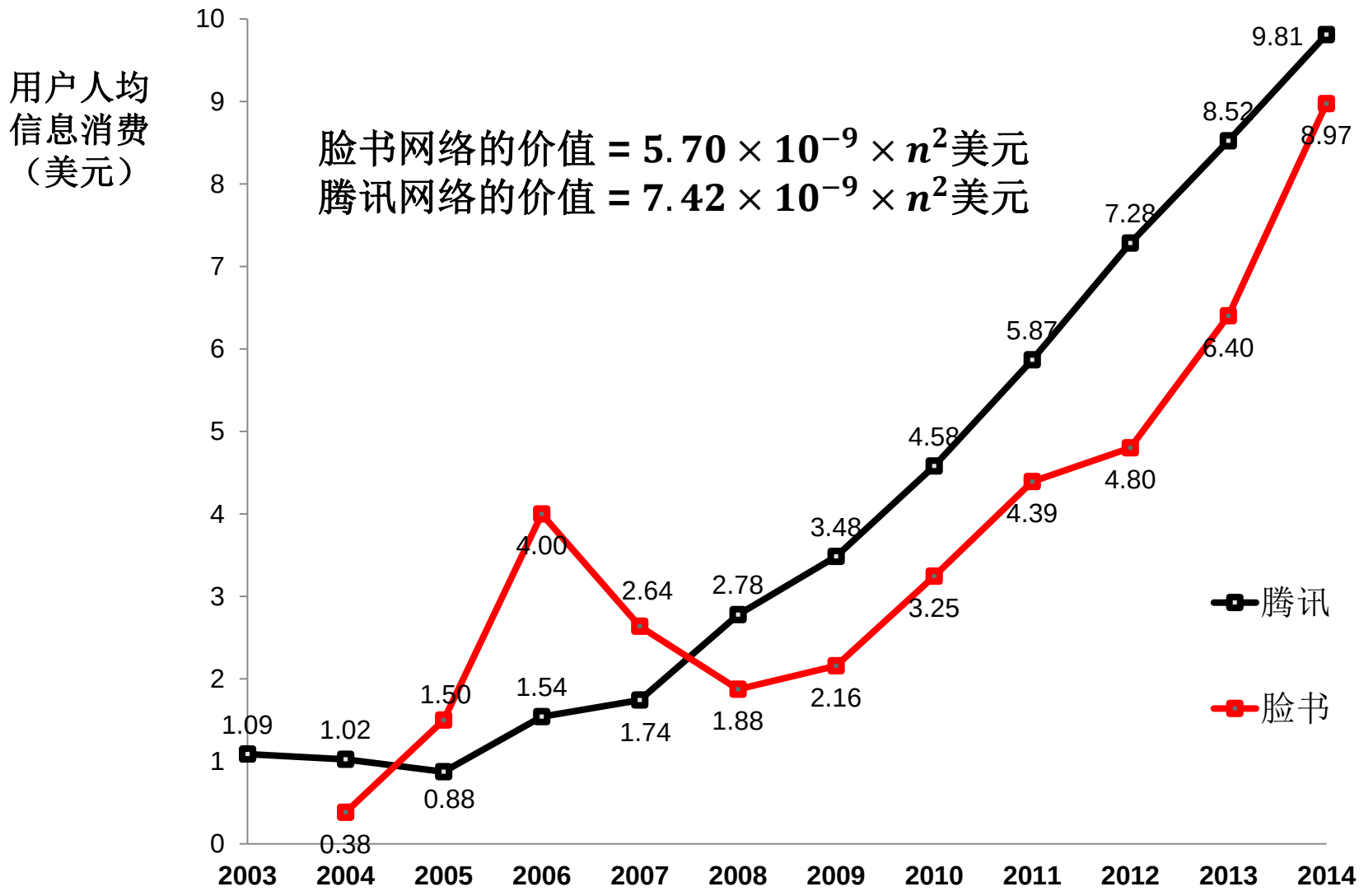
- 无为而治原理（the end-to-end argument）
  - 某些功能必须有边缘节点（应用）参与才能正确实现
    - 除非能够完全而正确地做到，网络不应该实现该功能
      - Keep it simple, stupid；我们不需要自作聪明的网络
- 网络效应：网络的价值随节点数超线性增长
  - 麦特考夫定律：网络的价值与节点数的平方成正比
  - 里德定律：网络价值＝2^{C-1}, C是“社区”个数
- 病毒性市场现象（viral marketing）
  - 低价格（购买成本为零）
  - 好使用（使用成本为零）
  - 易传播（传播成本为零）
  - 强黏糊（sticky）

E2E原理实例

- 为什么网络作业需要这么多查错？

```
 8  func main() {
 9      if httpresp, err := http.Get("http://csintro.ucas.ac.cn/static/code_project
         /Richard_Karp.txt"); err != nil || httpresp.StatusCode != http.StatusOK {
10              if err != nil {
11                      fmt.Fprintln(os.Stderr, err.Error())
12              } else {
13                      fmt.Fprintln(os.Stderr, httpresp.Status)
14              }
15              return
16      } else {
17              if data, err := ioutil.ReadAll(httpresp.Body); err != nil {
18                      fmt.Fprintln(os.Stderr, err.Error())
19              } else {
20                      fmt.Println(string(data))
21              }
22      }
23 }
```

脸书与腾讯数据验证了梅特卡夫定律



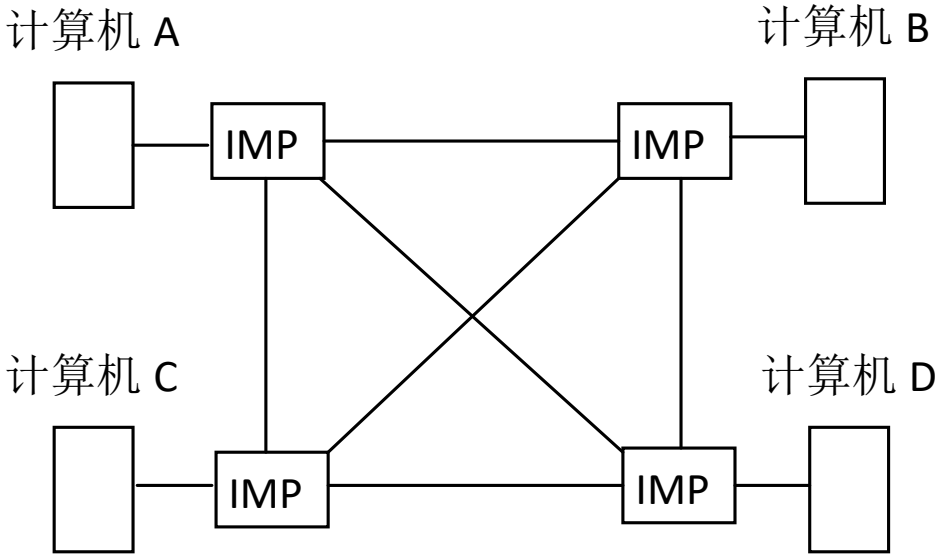
Zhang X, Liu J, Xu Z, Tencent and Facebook Data Validate Metcalfe's Law, JCST, 2015 Vol. 30 (2): 246-251.

电脑网络创新故事：ARPANET

- 条件与思想
  - 1957年10月4日，苏联成功地发射了人造卫星
  - 1958年，美国成立ARPA，IPTO
  - 1960年，利克莱德提出电脑网络思想
- 项目与实施
  - 1966年，鲍伯•泰勒启动ARPANET（为什么？）
  - 1966年秋，罗伯兹开始设计ARPANET
    - 1968年8月，罗伯兹完成了ARPANET的技术规范，并向全国140家公司发出了招标书
    - 1968年12月，BBN公司中标
    - 1969年9月，第一个ARPANET节点安装在加州大学洛杉矶分校

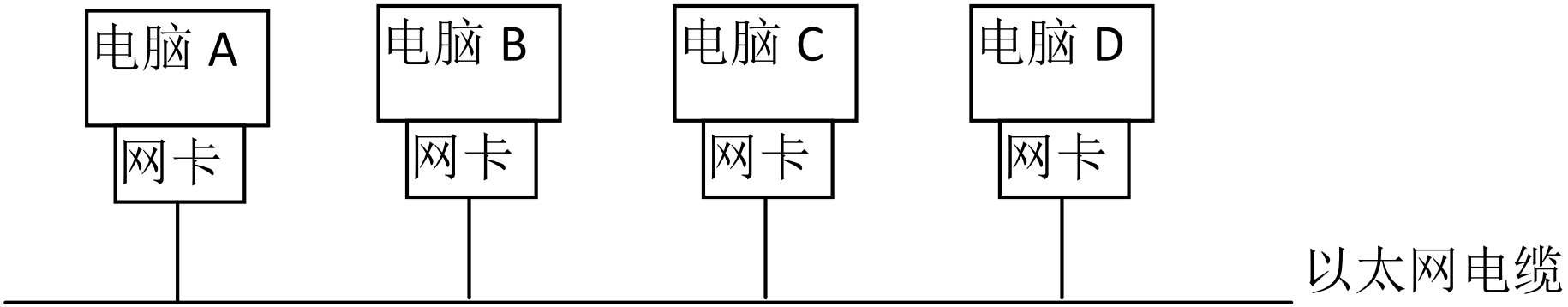
ARPANET成果

- 1969年10月29日，ARPANET进行了第一次试验，传输了“LO”GIN
- 第一个电脑网络
- 第一个计算机网络技术标准（RFC）
- 第一个交换机
- 重传容错机制
- 验证了分组交换
- **引发了三个新问题**
  - 局域网、路由器、因特网



局域网与以太网

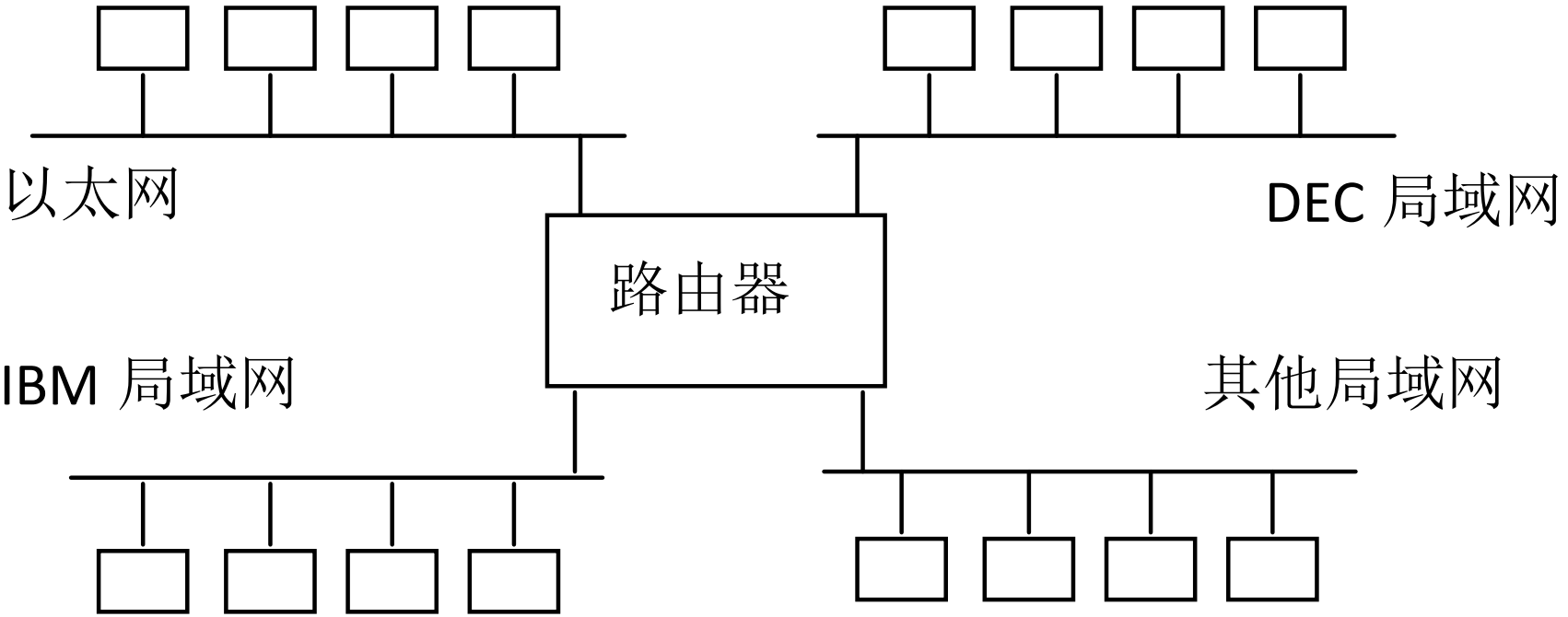
- 1973年，麦特考夫发明以太网
  - 网卡
  - 解决冲突的指数退避方法
    - 第一次传输试图失败后，等候[0, T]中间的一个随机值
    - 第二次重试失败后，等候[0, 2T]中间的一个随机值
    - 第三次重试失败后，等候[0, 4T]中间的一个随机值



以太网和四台电脑构成的局域网

校园网、城域网与路由器

- 1980-1983年，波沙克和勒纳尔实现路由技术
- 1983-1984年，波沙克和勒纳尔发明路由器
  - 专用设备



TCP/IP协议栈

- 1974年5月，康恩和舍夫在IEEE Transactions on Communication发表TCP/IP协议栈论文
  - 特点：协议栈
  - 因特网（Internet）技术正式诞生，它可以把很多计算机网络互联起来组成一个大的网络之网
- 1983年，ARPANET与美国国防部的另一个网络“国防数据网”开始使用TCP/IP协议
  - 有人把这个时间认为是因特网的真正诞生年代，因为我们今天所说的因特网，是指使用IP的网络之网

早期应用：主要并不是计算资源共享

- FTP、Telnet、BBS、电子邮件
- 1972年，BBN工程师汤姆林生在快完成FTP的编程工作时突发奇想：为什么不能用FTP来自动地传送网络电子邮件呢?
  - 本机电子邮件技术 + 远程文件传输技术FTP
  - mailto: zxu@ict.ac.cn
- 1973年，ARPANET上的四分之三的通信是电子邮件
- 电脑网络就只是一个更快的邮局系统吗？

万维网

- 1980-1990年，提姆•伯纳尔斯-李发明万维网
  - 核心思想和价值：将超文本、超链接技术从单机内拓展到全球计算机网络，实现全球电脑网络中的文档资源互连
  - 四项关键技术
    - 文档资源的命名：URL
    - 文档资源的表示：HTML
    - 文档资源的访问：HTTP
    - 核心软件：Web服务器与浏览器
- 1993年，安德雷生发明马赛克Web浏览器
  - 1994年成为网景浏览器（Netscape）

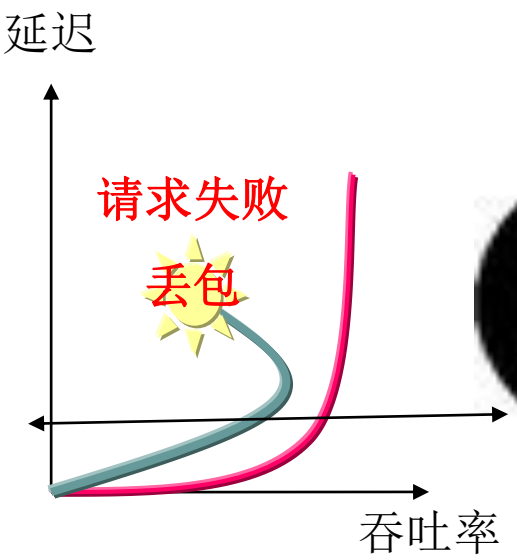
社交网络

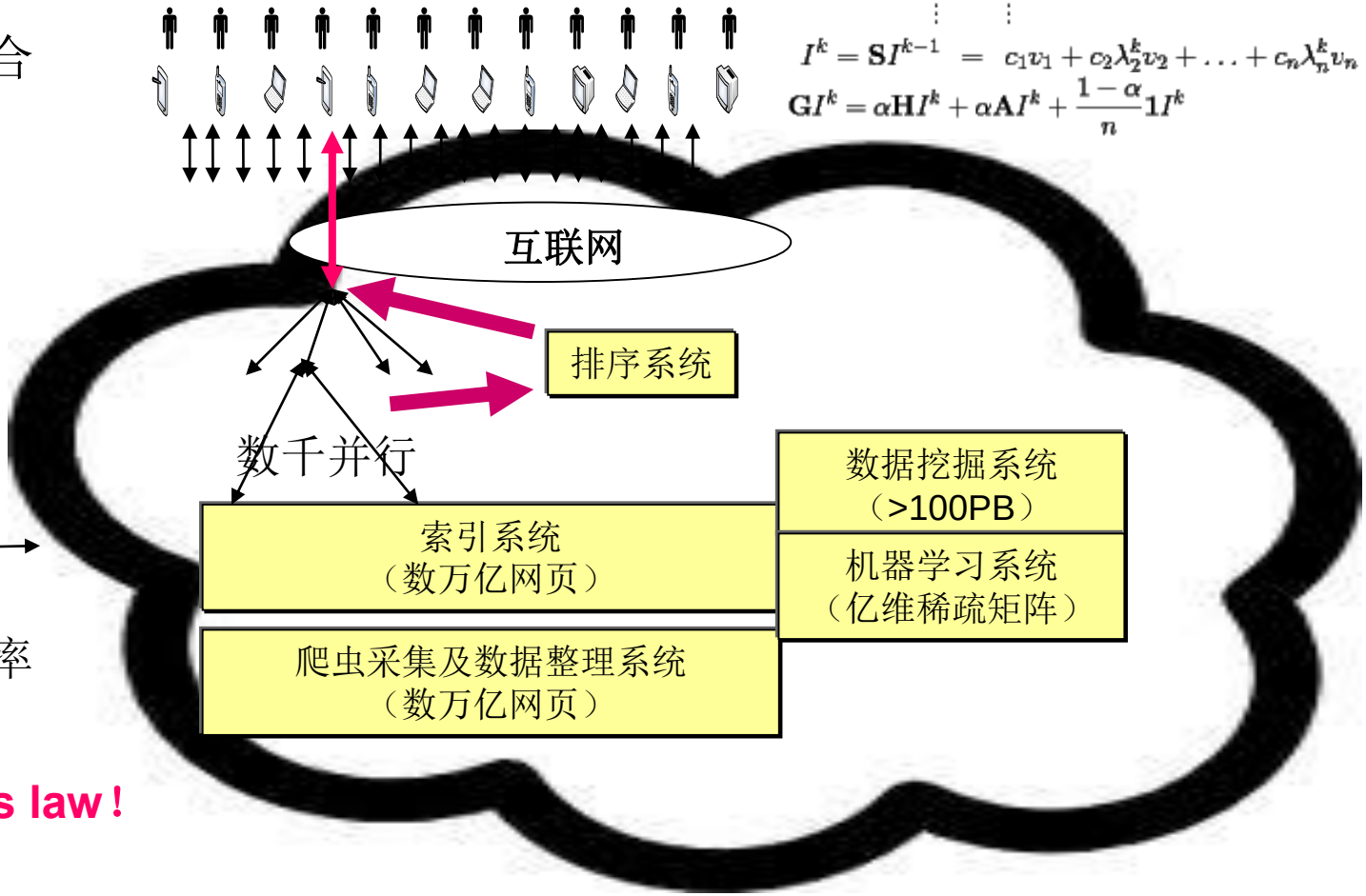
- 2003年11月19日，Harvard Crimson
  - 大学管委会决定放二年级学生马克•扎克伯格（Mark Zuckerberg）一马，不予他离校处分
  - Facemash选“辣的”同学，未经允许使用同学照片，涉嫌破坏计算机网络安全、侵犯版权和侵犯个人隐私
- 2004年2月4日，脸书网（也就是今天的Facebook社交网络服务）正式上线
  - 扎克伯格可能已经满足了“**1万小时定律**”
- 2012年，脸书公司在纳斯达克股票市场上市，同年用户数增长到9亿人；2014年14亿人
- 与MySpace不同，密切参加开源社区（如Hive）

互联网搜索业务系统：在线+离线计算

- 在线性能指标：Amazon三元组
  (最大请求数,质量百分位,响应时间)
  =(百万, 99.9%, 10微秒）
- 离线指标：一天
- 两种模式相互配合

$$I^0 = c_1 v_1 + c_2 v_2 + \ldots + c_n v_n$$
$$I^1 = \mathbf{S}I^0 = c_1 v_1 + c_2 \lambda_2 v_2 + \ldots + c_n \lambda_n v_n$$
$$I^2 = \mathbf{S}I^1 = c_1 v_1 + c_2 \lambda_2^2 v_2 + \ldots + c_n \lambda_n^2 v_n$$
$$\vdots \quad \vdots$$
$$I^k = \mathbf{S}I^{k-1} = c_1 v_1 + c_2 \lambda_2^k v_2 + \ldots + c_n \lambda_n^k v_n$$
$$\mathbf{G}I^k = \alpha \mathbf{H} I^k + \alpha \mathbf{A} I^k + \frac{1-\alpha}{n} \mathbf{1} I^k$$



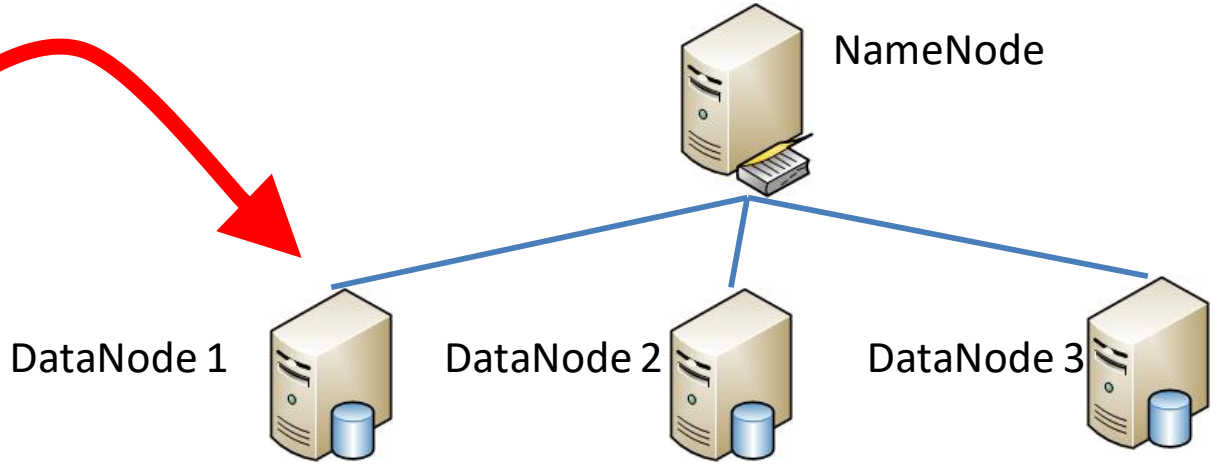


实际情况差于Little's law！

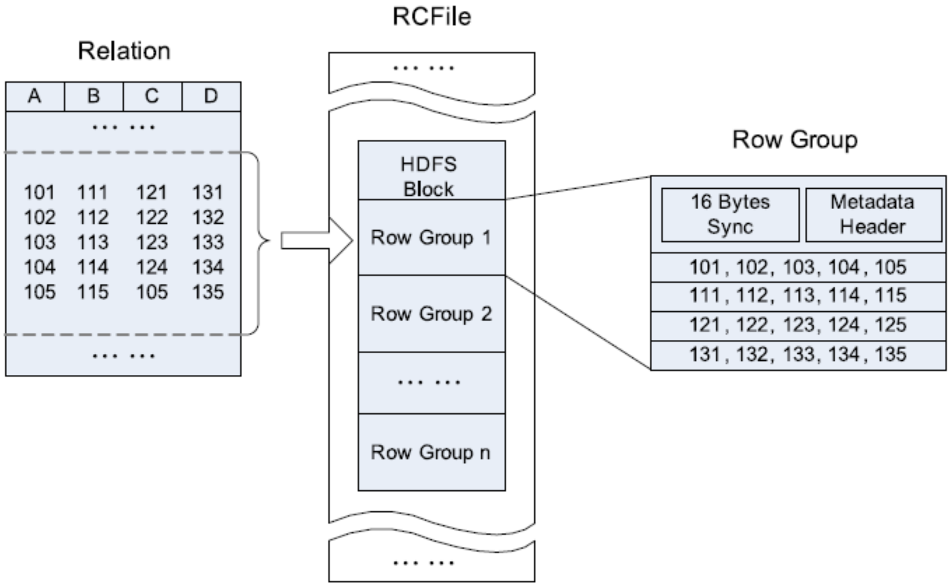
网络计算系统的一个实例：RCFile

- 大数据计算的数据放置问题
  - 如何将100-1000PB的 数据放置到数千个计算结点中，使得离线数据挖掘计算所需存储空间最小、速度最快？

A	B	C	D
101	201	301	401
102	202	302	402
103	203	303	403
104	204	304	404
105	205	305	405

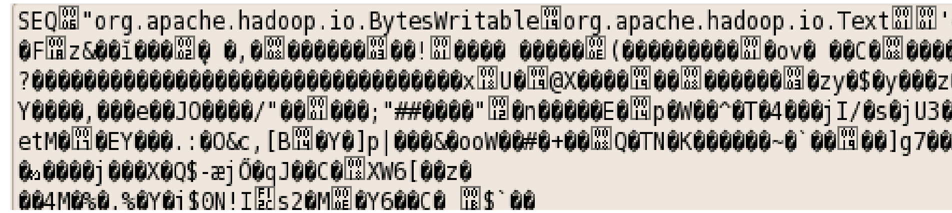


- 2010年，中科院计算所何永强同学
  - 发明行列混合存储技术RCFile
  - 开源到Apache Hive社区，全球使用
  - **http://en.wikipedia.org/wiki/RCFile**
- 2016年何永强在干什么？



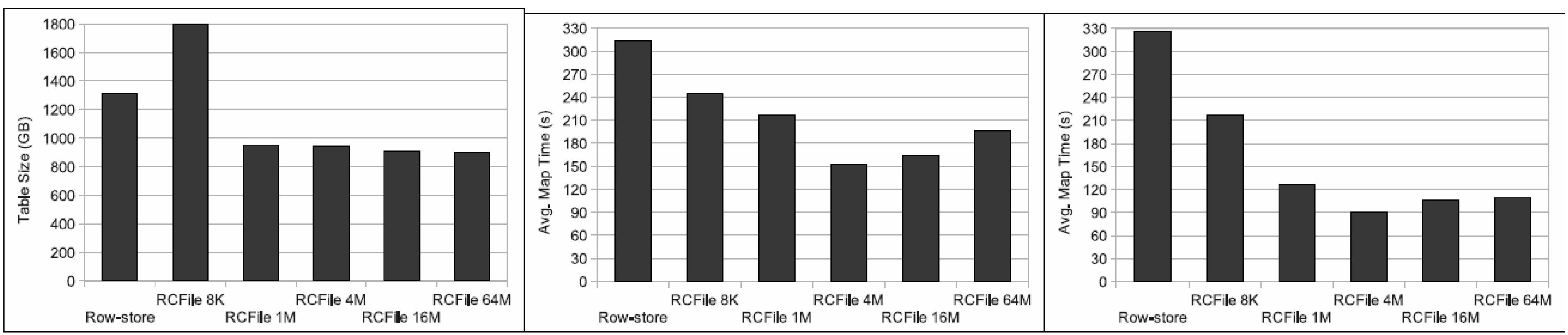
RCFile data layout in HDFS blocks
(Relation partitioned into row groups, each group column stored)

Plain text format: 10.11MB



Compressed row-store format: 2.13MB

RCFile format: 1.8MB

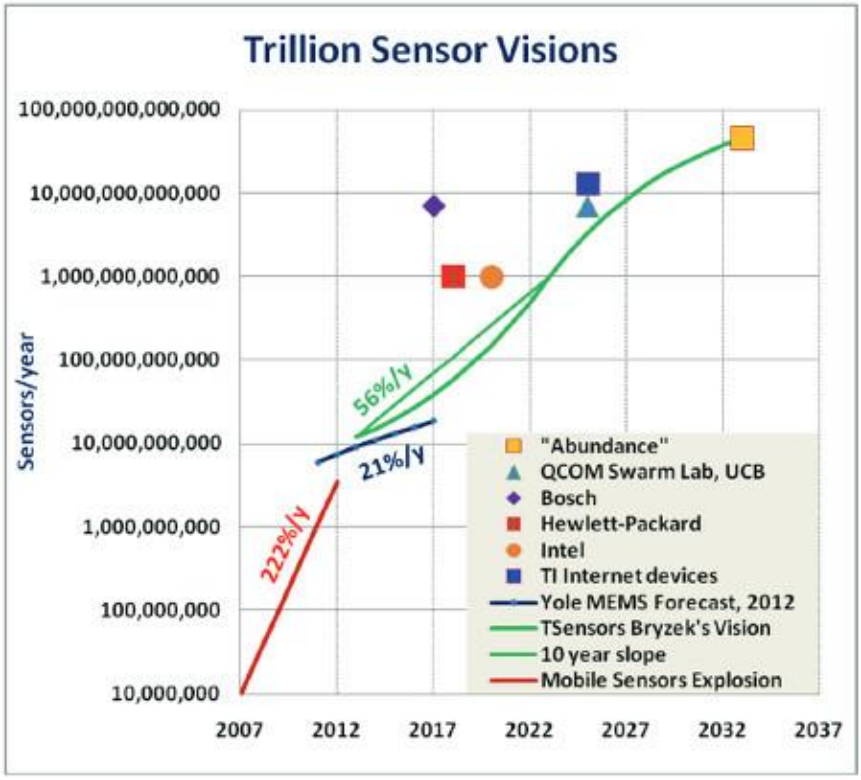


(a) Table Size (b) Query A (c) Query B

Testing results on Facebook workload

未来十五年信息技术发展趋势研判

- 继桌面互联网、移动互联网之后，信息技术正在进入第三个宏观阶段，万物互联网成为重要方向
  - 互联网→移动互联网→万物互联网（IoE）
  - **IoE: Internet of Everything**
    - **Everything = People + Data + Processes + Things**
- 多个预测：
  - 2030年，全球将有千亿~万亿传感器，数百亿物端设备；每个设备都有处理器、OS、开发环境、使用模式



物端产业生态尚未变成红海

- 产业界：很多小批量产品与方案；无主流生态
- 昆虫纲悖论

谢谢 Thank You

Q&A

zxu@ict.ac.cn

中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY