

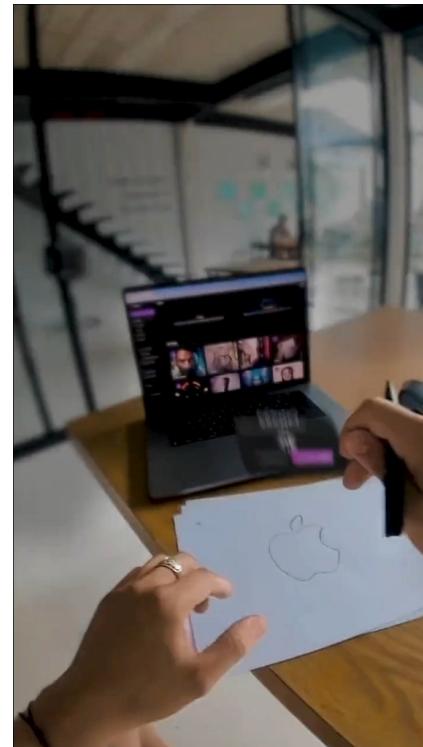
# Introduction to CSE: Coping with Complexity

Yubin Xia

IPADS, Shanghai Jiao Tong University

<https://www.sjtu.edu.cn>

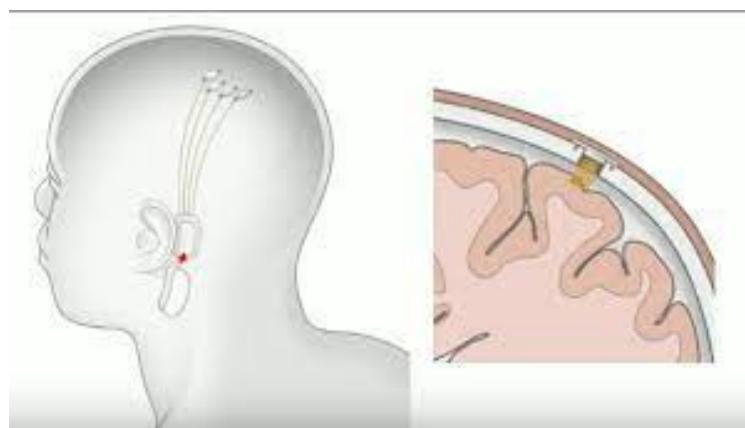
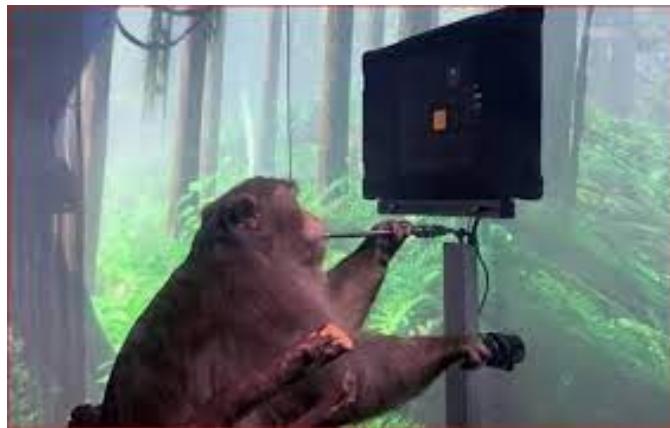
# Digital-world VS. Real-world



# Digital-world VS. Real-world



# Human Computer Interface...



**Windows**

A fatal exception 0E has occurred at 0028:C0011E36 in VXD UMM(01) + 00010E36. The current application will be terminated.

- \* Press any key to terminate the current application.
- \* Press CTRL+ALT+DEL again to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue \_

# 钉钉回应黑屏、崩溃 Bug：扛到最后，没想到还是摔了一跤

2020年03月10日 12:05 IT之家



新浪财经APP

A<sup>-</sup>

A<sup>+</sup>

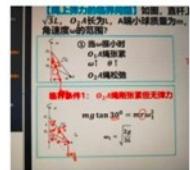
☆



也不知道该取个啥名儿

今天钉钉是不是有点问题？？？

英语课的时候发的消息在物理课又蹦出来了😭😭还有突然黑屏，简直了



47秒前 来自 OPPO K3 硬核少年

收藏

转发

评论



郭雅20416

上着上着，部分同学说黑屏了，退出重进也没用，只好关了直播重新开，还以为是我自己电脑或网络有问题，原来是钉钉自己终于承受不住压力崩了😭

@圈内芒果捞

#钉钉崩了#

钉钉终于承受不住大家沉重的爱，崩了？

有多人今天快乐了 😊



+关注

按热度 | 按时间

思南执北 V: 在？？你崩了

今天 09:34

回复 | 1995

吃薯片的橙子 等人 共68条回复

宁陪你看那细水长流: 钉钉的未成年用户太多了，建议加上防沉迷系统，两小时以后强制下线  
3月9日 21:31

回复 | 1142

醉卧庐浅笑 等人 共16条回复

捡破烂的周树人: 在？？？你好像崩了？？？  
今天 09:24

回复 | 836

青春往昔浮年 等人 共4条回复

1128号飞船: 你崩了！@钉钉  
今天 09:43

回复 | 534

橙子味的鹤归 等人 共4条回复

大外昆仑废青年: 你这只呆鸟 崩了 😂  
今天 10:26

回复 | 464

# 5·27支付宝大规模宕机事故

语音

编辑

讨论

+ 上传视频

2015年5月27日下午，部分用户反映其支付宝出现[网络故障](#)，[账号](#)无法登录或支付。[支付宝](#)官方表示，该故障是由于[杭州市萧山区](#)某地光纤被挖断导致，这一事件造成部分用户无法使用支付宝。随后支付宝工程师紧急将用户请求切换至其他机房，受影响的用户逐步恢复。到晚上7点20分，支付宝方面宣布用户服务已经恢复正常。

中文名	5·27支付宝大规模宕机事故	恢复时间	2015年5月27日20时左右
发生时间	2015年5月27日下午4点半左右	事件影响	全国大面积支付宝瘫痪

## 目录

- [1 事故背景](#)
- [2 事故经过](#)

- [3 事故原因](#)
- [4 事故处理](#)

- [5 事故影响](#)
- [6 各方评价](#)

## 事故背景

语音

编辑

[支付宝](#)拥有超过4万亿年交易总额，是中国第一大第三方交易平台，约占中国整体社会消费金额的六分之一。

2014年11月，就有用户反映，支付宝钱包目前无法转账和提现，当用户使用这两项功能时会提示出现未知错误或创建交易失败，该问题在移动客户端以及电脑网页端均存在。[\[1\]](#)

PATIENT NAME: John  
TREATMENT MODE: FIX

UNIT RATE/MINUTE  
MONITOR UNITS  
TIME (MIN)

Therac-25(radiation therapy machine ) accidents (1985-1987), the injured patients died from lethal dosage of radiation caused by a software bug

	ACTUAL	PRESCRIBED	
GANTRY ROTATION (DEG)	0.000000	0.000000	VERIFIED
COLLIMATOR ROTATION (DEG)	359.200000	359.200000	VERIFIED
COLLIMATOR X (CM)	14.200000	14.200000	VERIFIED
COLLIMATOR Y (CM)	27.200000	27.200000	VERIFIED
WEDGE NUMBER	1.000000	1.000000	VERIFIED
ACCESSORY NUMBER	0.000000	0.000000	VERIFIED

DATE: 2012-04-16  
TIME: 11:48:58  
OPR ID: 033-tfs3p

SYSTEM: BEAM READY  
TREAT: TREAT PAUSE  
REASON: OPERATOR

OP.MODE: TREAT  
X-RAY  
COMMAND: █

AUTO  
173777

A blurry, grainy photograph of a rocket launching. The rocket is mostly white with some dark stripes. It has just taken off and is positioned vertically in the center of the frame. A massive, bright orange and yellow fireball or smoke plume is erupting from its base, filling the lower half of the image. The background is a hazy grey-blue.

Ariane 5 explosion due  
to a floating point bug



Jeep 自由光系统存在入侵风险，菲亚特克莱斯勒汽车公司召回 140 万辆汽车



# **SYSTEM & COMPLEXITY**

# What is a System?

**System =**

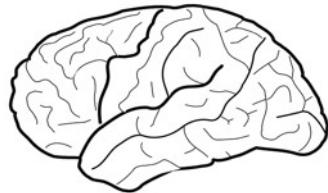
- Interacting set of components with a specified behavior at the interface with its environment

**Examples**

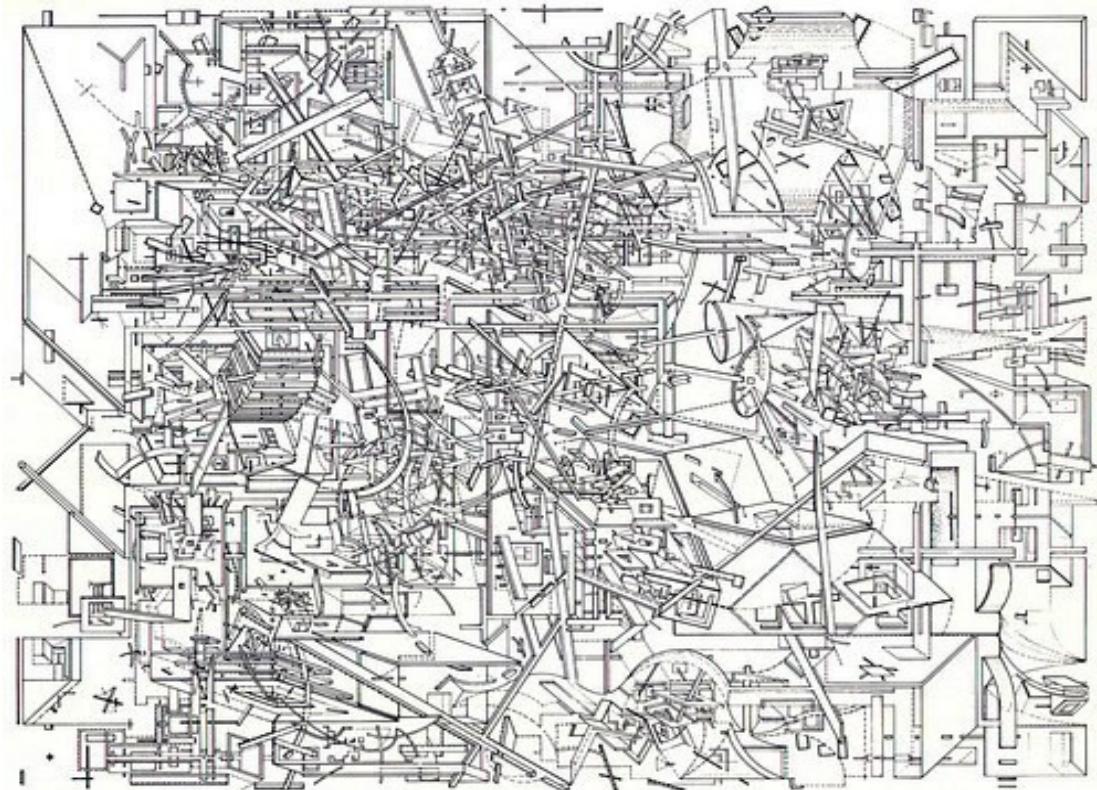
- A web server, an Android phone, file systems, Linux, ...

# The Problem: Complexity of the Systems

Hard for human to fully understand



But, we can view from a system's perspective



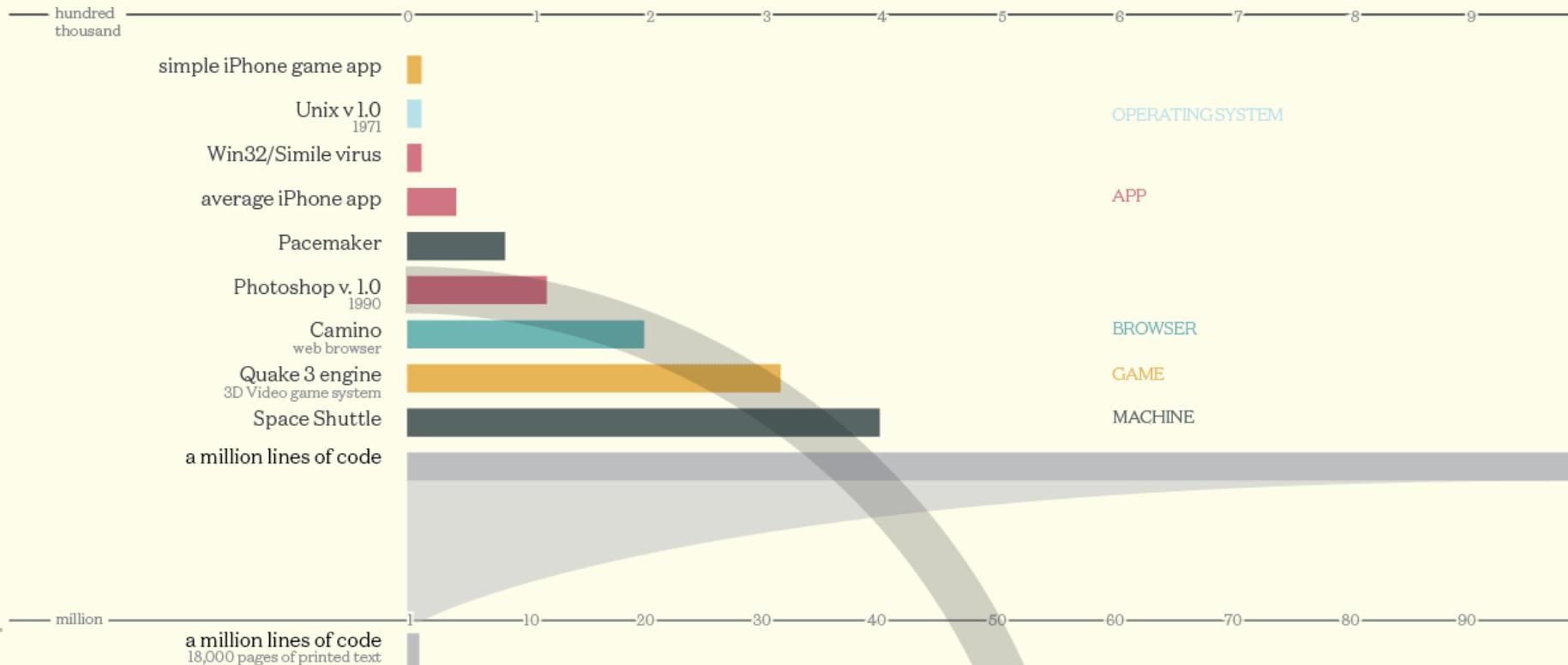
# Example of Complex Systems: Linux Kernel

Item	Lines	%
./usr	845	0.0042
./init	5,739	0.0283
./samples	8,758	0.0432
./ipc	8,926	0.0440
./virt	10,701	0.0527
./block	37,845	0.1865
./security	74,844	0.3688
./crypto	90,327	0.4451
./scripts	91,474	0.4507
./lib	109,466	0.5394
./mm	110,035	0.5422
./firmware	129,084	0.6361
./tools	232,123	1.1438
./kernel	246,369	1.2140
./Documentation	569,944	2.8085
./include	715,349	3.5250
./sound	886,892	4.3703
./net	899,167	4.4307
./fs	1,179,220	5.8107
./arch	3,398,176	16.7449
./drivers	11,488,536	56.6110

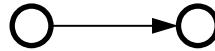
# LOC (Lines Of Code)

## Codebases

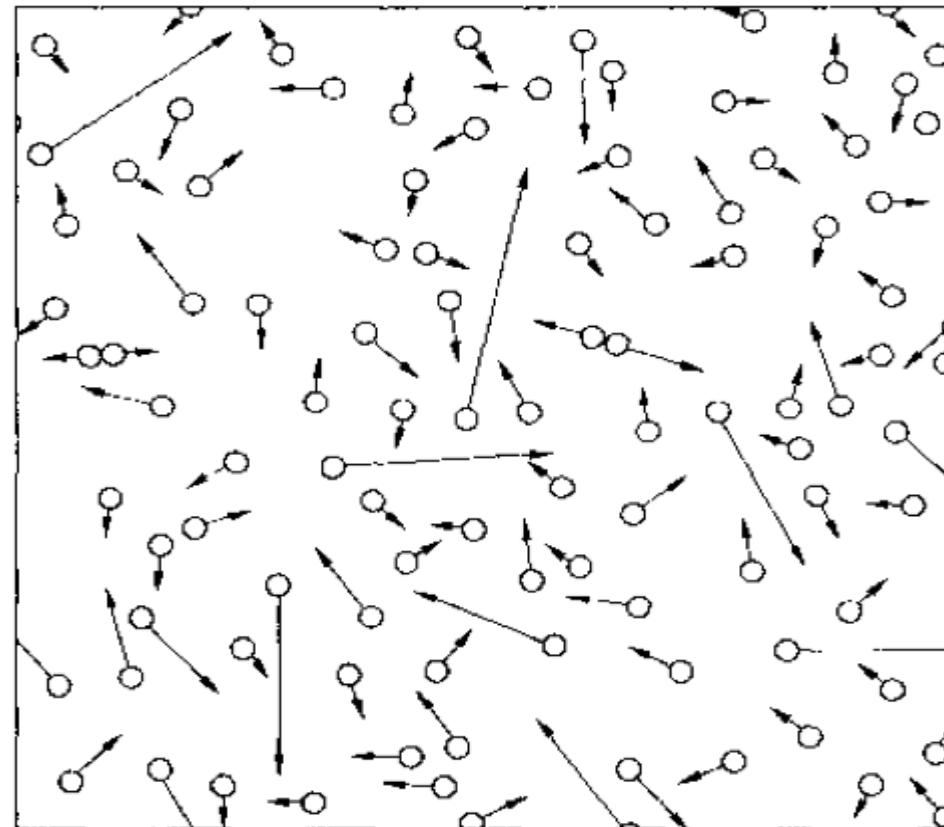
Millions of lines of code



# An Example: The Gas System



A systematical way to understand a complex system is needed



# Compare with the Computer Systems

## Programming / Data Structure

- LoC (Lines of Code): From hundreds to millions

## Operating System / Architecture

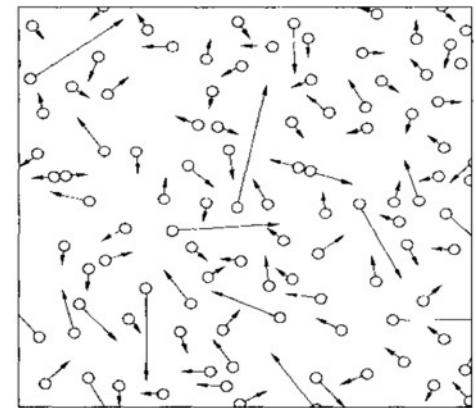
- CPU cores: from one to hundreds

## Network

- Nodes: from two to millions

## Web Service

- Clients: from tens to millions





# **14 PROPERTIES**

# 14 Properties of Computer Systems

- 1. Correctness
- 2. Latency
- 3. Throughput
- 4. Scalability
- 5. Utilization
- 6. Performance Isolation
- 7. Energy Efficiency
- 8. Consistency
- 9. Fault Tolerance
- 10. Security
- 11. Privacy
- 12. Trust
- 13. Compatibility
- 14. Usability

# Property-1: Correctness

## Correctness is hard to define

- How to describe the functionalities of your code?

## 'It's Not a Bug, It's a Feature.'

Who can really say? In a 2013 study, a group of scholars at a German university sifted through the records of five software projects and evaluated thousands of reported coding errors. They discovered that the bug reports were themselves thoroughly buggy. “Every third bug is not a bug,” they concluded. The title of their paper will surprise no one: “It’s Not a Bug, It’s a Feature.”

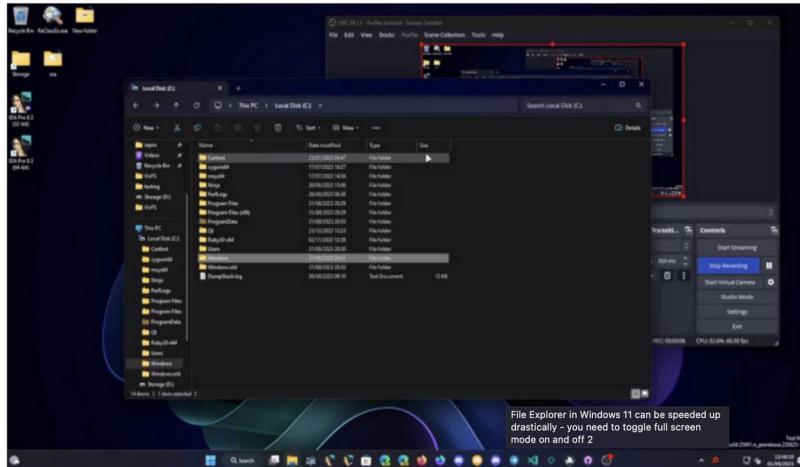
<https://www.wired.com/story/its-not-a-bug-its-a-feature/>

# Property-1: Correctness

**File Explorer in Windows 11 can be speeded up drastically – you need to toggle full screen mode on and off**

6 days ago · Robbie Elmers · Add Comment

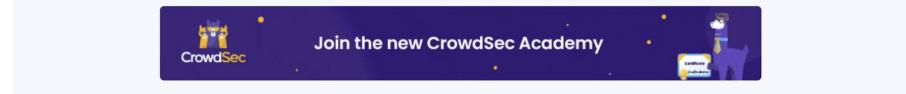
File Explorer in Windows 11 is not particularly fast, but thanks to an unknown feature of the program, a way was found to speed it up drastically – you need to enable and disable full screen mode. This method was reported by the user of the social network X under the nickname [Vivi](#).



## The Hacker News

Home Data Breaches Cyber Attacks Vulnerabilities Webinars Store Contact

Get the Free Newsletter



You can Hack into a Linux Computer just by pressing 'Backspace' 28 times

Dec 17, 2015 · Swati Khandelwal



## A Piece of Code, Seems Normal

```
char *buf = ...;
char *buf_end = ....;
unsigned int len = ....;
if (buf + len >= buf_end)
    return; /* len too large */
if (buf + len < buf)
    return; /* overflow, buf+len wrapped around */
/* write to buf[0..len-1] */
```

---

# Undefined Behavior

	Construct	Sufficient condition	Undefined behavior
Language	$p + x$	$p_\infty + x_\infty \notin [0, 2^n - 1]$	pointer overflow
	$*p$	$p = \text{NULL}$	null pointer dereference
	$x \text{ op}_s y$	$x_\infty \text{ op}_s y_\infty \notin [-2^{n-1}, 2^{n-1} - 1]$	signed integer overflow
	$x / y, x \% y$	$y = 0$	division by zero
	$x \ll y, x \gg y$	$y < 0 \vee y \geq n$	oversized shift
	$a[x]$	$x < 0 \vee x \geq \text{ARRAY\_SIZE}(a)$	buffer overflow
Library	$\text{abs}(x)$	$x = -2^{n-1}$	absolute value overflow
	$\text{memcpy(dst, src, len)}$	$ \text{dst} - \text{src}  < \text{len}$	overlapping memory copy
	use $q$ after $\text{free}(p)$	$\text{alias}(p, q)$	use after free
	use $q$ after $p' := \text{realloc}(p, \dots)$	$\text{alias}(p, q) \wedge p' \neq \text{NULL}$	use after realloc

# Is This Compilers' Bug?

A true story: GCC bug #30475 (2007/01/15)

A GCC user:

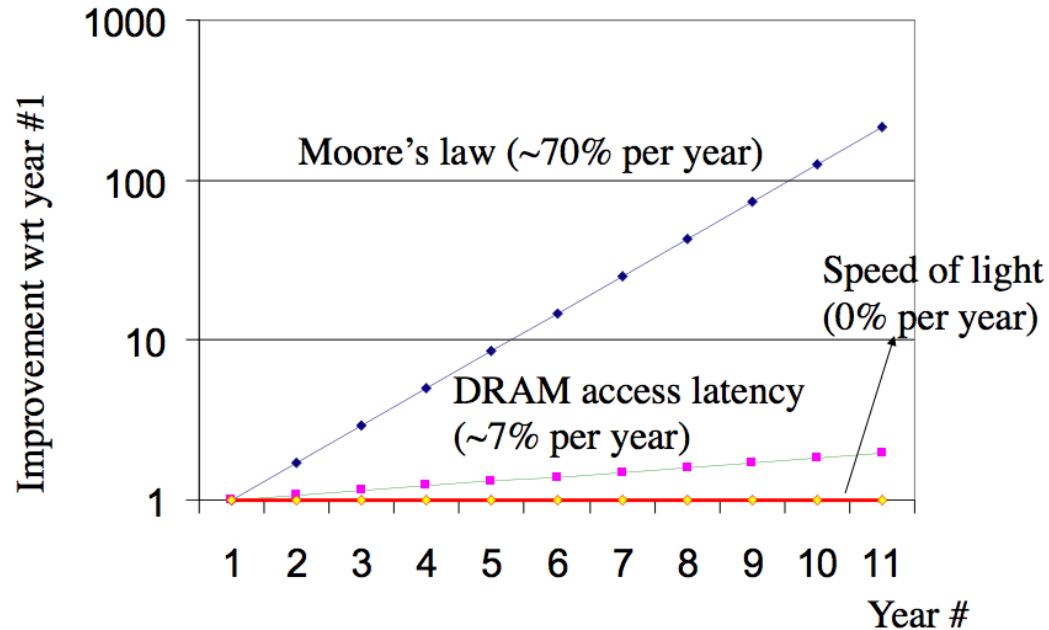
- *"This will create MAJOR SECURITY ISSUES in ALL MANNER OF CODE. I don't care if your language lawyers tell you gcc is right. . . . FIX THIS! NOW!"*

A GCC developer:

- *"I am not joking, the C standard explicitly says signed integer overflow is undefined behavior. . . . GCC is not going to change."*

## Property-2: Latency

Latency is hard to optimize, it has physical limitations



## Property-3: Throughput / Capacity

Throughput: a number of transactions / transferred-data per second

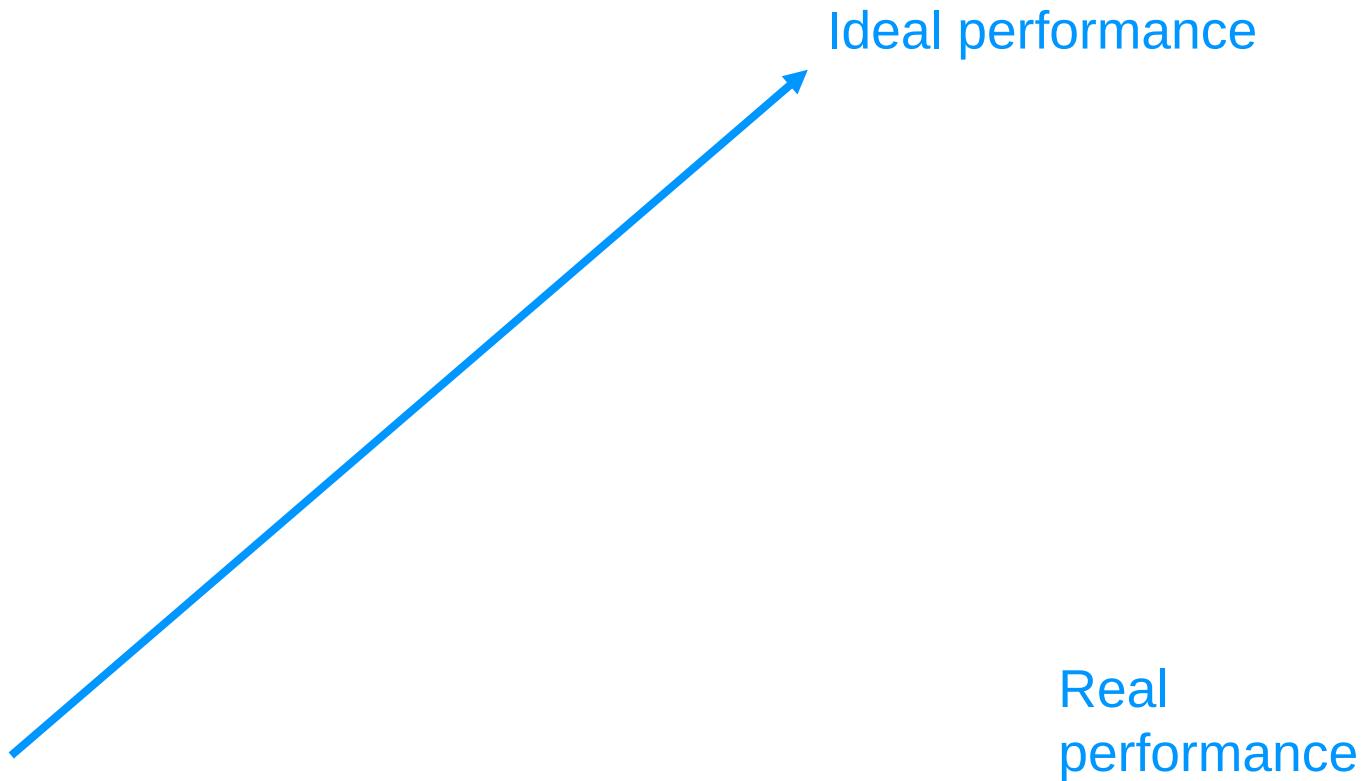
7nm → 5nm → 3nm → 2nm... what's next?

Once stop getting smaller, it has to go bigger

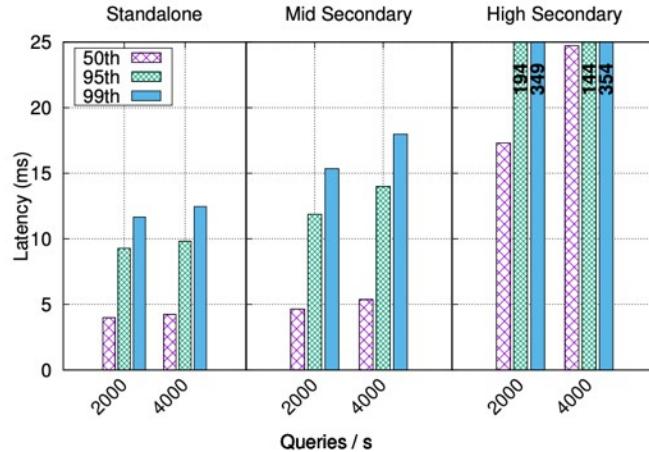
### **CES 2019: Moore's Law is dead, says Nvidia's CEO**

The long-held notion that the processing power of computers increases exponentially every couple of years has hit its limit, according to Jensen Huang.

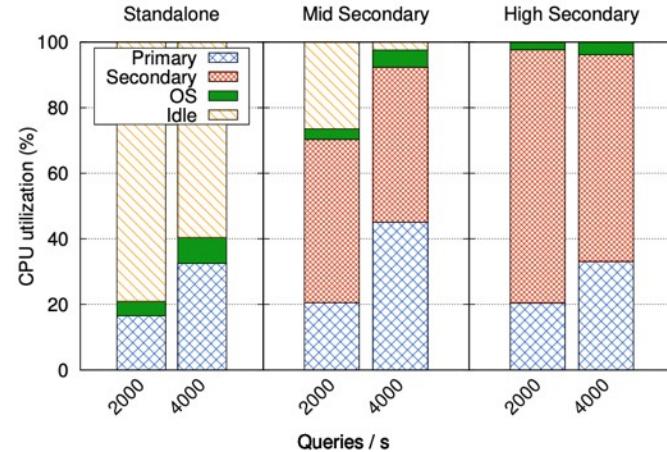
## Property-4: Scalability



# Property-5: Performance Isolation



(a) Query response latency



(b) CPU utilization

Figure 4: Single machine run of IndexServe standalone (no colocation) vs. colocated with an unrestricted secondary. A *mid* secondary increases the 99<sup>th</sup> percentile query latency by up to 42%, while a *high* increases same by up to 29×.

# Property-6: Utilization

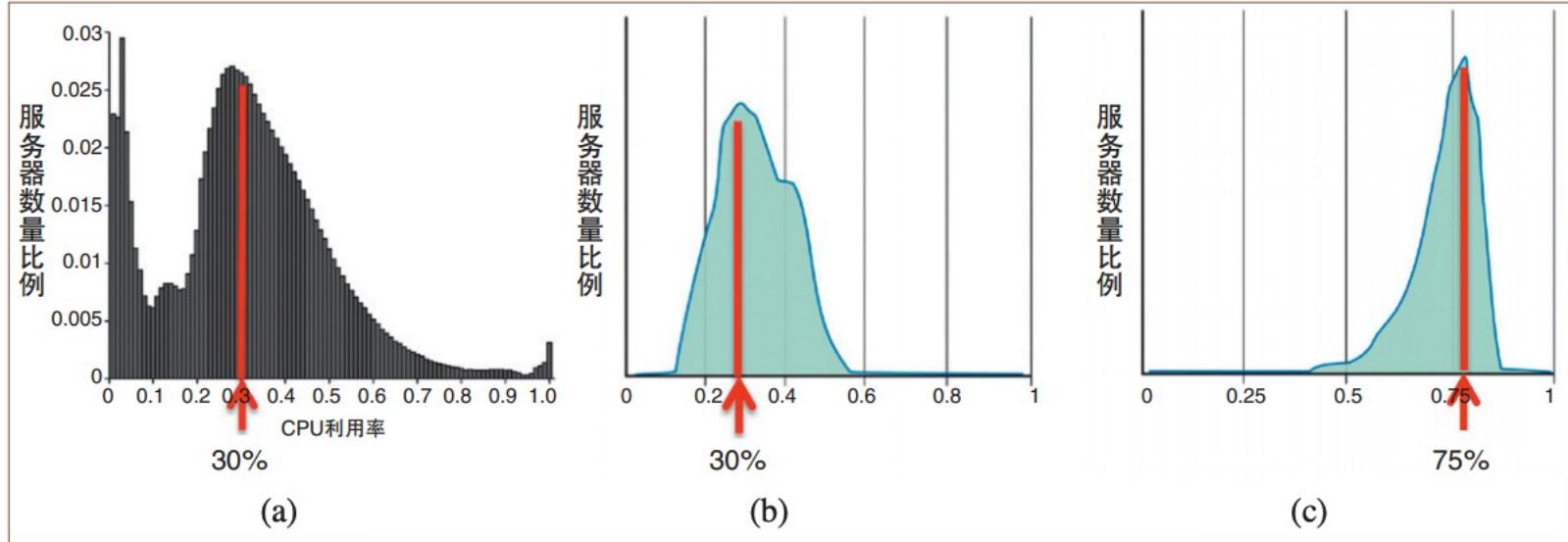
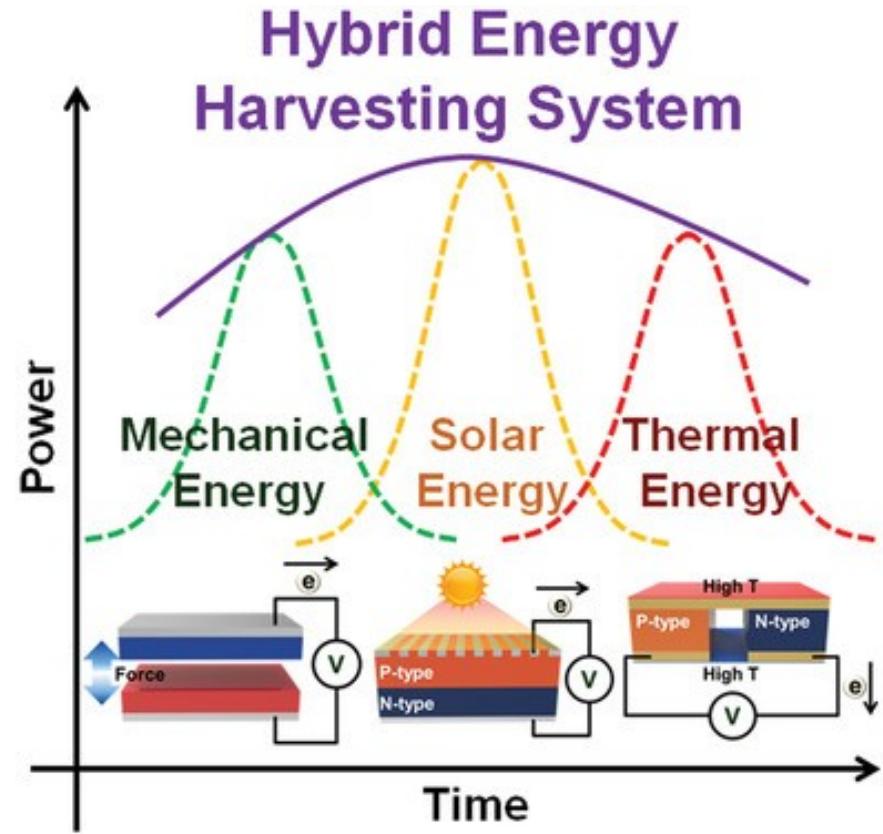


图2 谷歌数据中心CPU利用率: (a)2006年5000台在线应用服务器; (b)2013年2万台在线应用服务器;  
(c)2013年2万台批处理应用服务器<sup>4</sup>

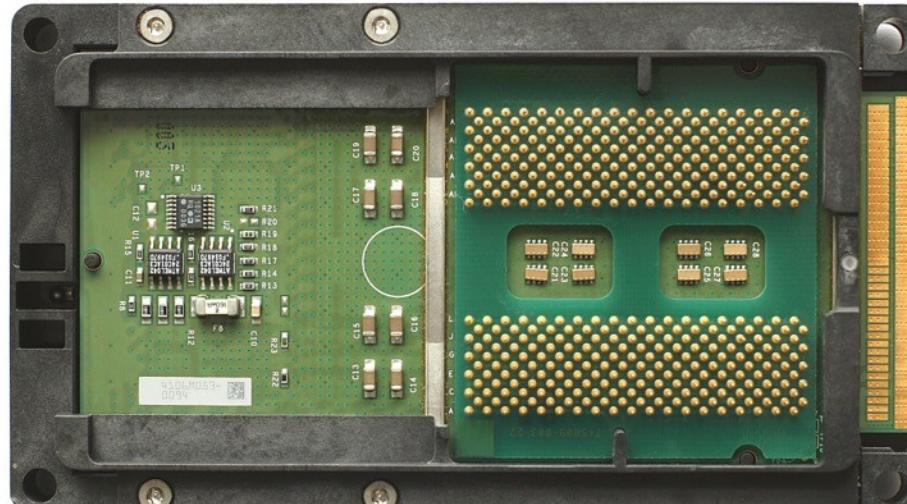
数据来源：“数据中心成本与利用率现状分析”，中国计算机学会通讯，2015.

# Property-7: Energy Efficiency



# Property-8: Compatibility

Intel's Itanium, 64-bit, not compatible to x86, died.



Itanium Process, from Wikipedia

# Property-9: Usability



# Property-10: Consistency

## 在12306买到相同高铁票怎么办？

2019-06-10 17:12

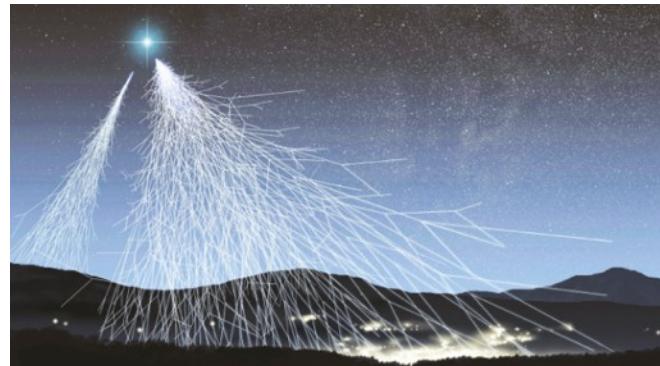
据时间视频消息，6月8日，北京南站，庞女士和丈夫乘坐G497列车，上车后发现票面上的座位坐了人。庞女士和在座位上的乘客交涉后，发现除了身份证件和姓名不一样之外，她们的票和自己手上的票一模一样。两人均是在窗口购票，双方购票相差时间三分钟左右。双方在找到列车长之后，庞女士被安排到了另一个车厢。

针对此事，12306客服回应称这种情况很少见，可能是由于机器故障所致。



# Property-11: Fault Tolerance

A bug of Cisco router is caused by cosmic radiation



Cisco Bug: CSCuz62750 - Incremental drops on counter DROP\_FRM\_CRC\_ERR\_SGMII0 causes traffic loss

**Last Modified**

Sep 20, 2016

**Product**

Cisco ASR 9000 Series Aggregation Services Routers

**Known Affected Releases**

all

**Description** (partial)

**Symptom:**

Partial data traffic loss can be observed. Below list counters need to focus to determine the issue:-

"show controller np counters all loc <location>" could indicate FRM\_FRM or CRC drop counts.

1082	DROP_FRM_CRC_ERR_SGMII0	1429516994	0
1083	DROP_FRM_FRM_ERR_SGMII0	74455703	0
1084	DROP_FRM_CRC_ERR_SGMII1	1427728248	0
1085	DROP_FRM_FRM_ERR_SGMII1	74942820	0
1086	DROP_FRM_CRC_ERR_SGMII2	1431171380	0
1087	DROP_FRM_FRM_ERR_SGMII2		
1088	DROP_FRM_CRC_ERR_SGMII3		
1089	DROP_FRM_FRM_ERR_SGMII3		

**Cosmic radiation**

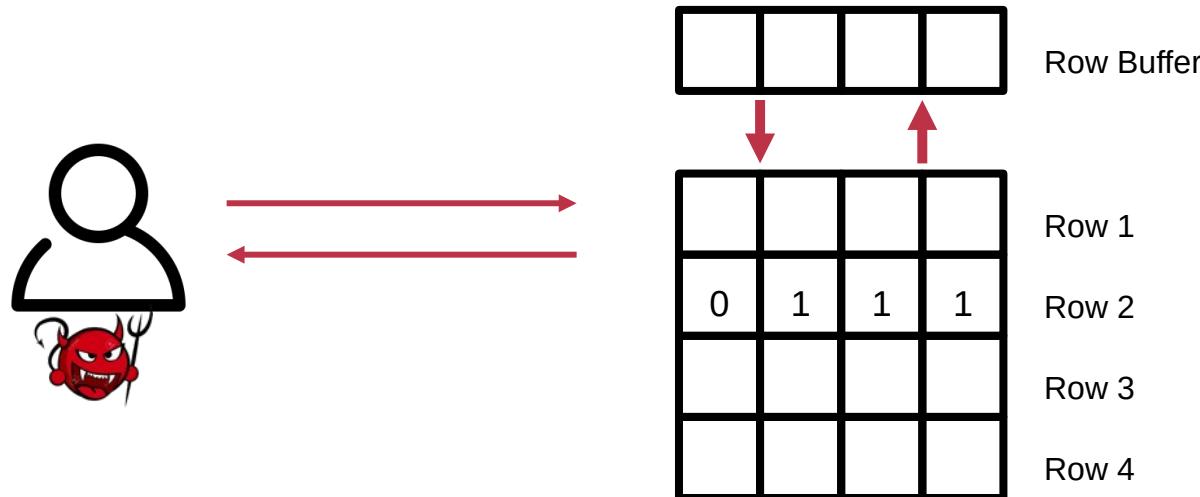
It is also theoretically possible for data corruption to happen after the CRC check, which should then be detected at the higher protocol layers.

**Conditions:**

Problem observed on operational network. Possible trigger is cosmic radiation causing SEU soft errors.

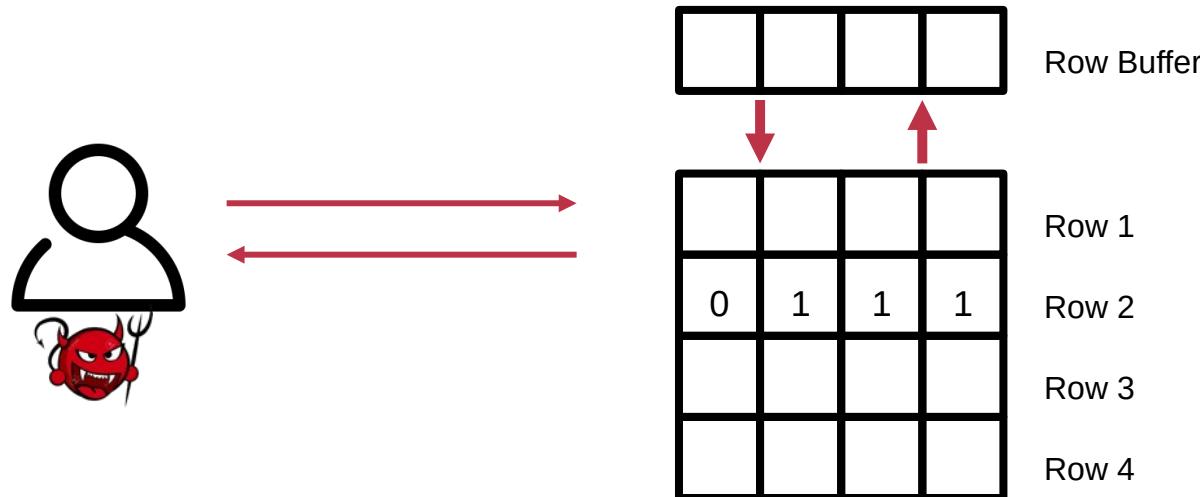
This issue can affect "Juggernaut" LCs (A9K-1X100GE-SE/TR, A9K-2X100GE-SE/TR)

# Property-11: Fault Tolerance



Rowhammer Attack: flip memory bits!

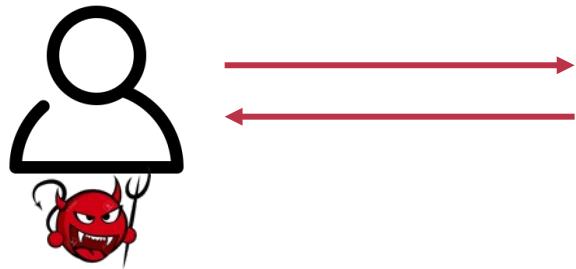
# Property-11: Fault Tolerance



Rowhammer Attack: flip memory bits!

# Property-11: Fault Tolerance

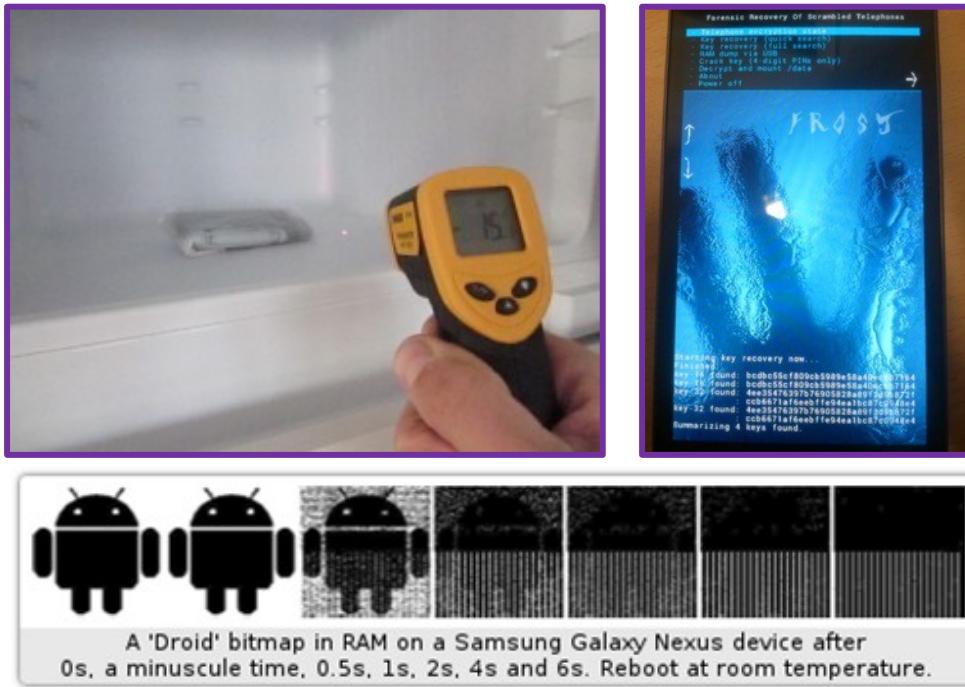
CPU Architecture	Errors	Access-Rate
Intel Haswell (2013)	22.9K	12.3M/sec
Intel Ivy Bridge (2012)	20.7K	11.7M/sec
Intel Sandy Bridge (2011)	16.1K	11.6M/sec
AMD Piledriver (2012)	59	6.1M/sec



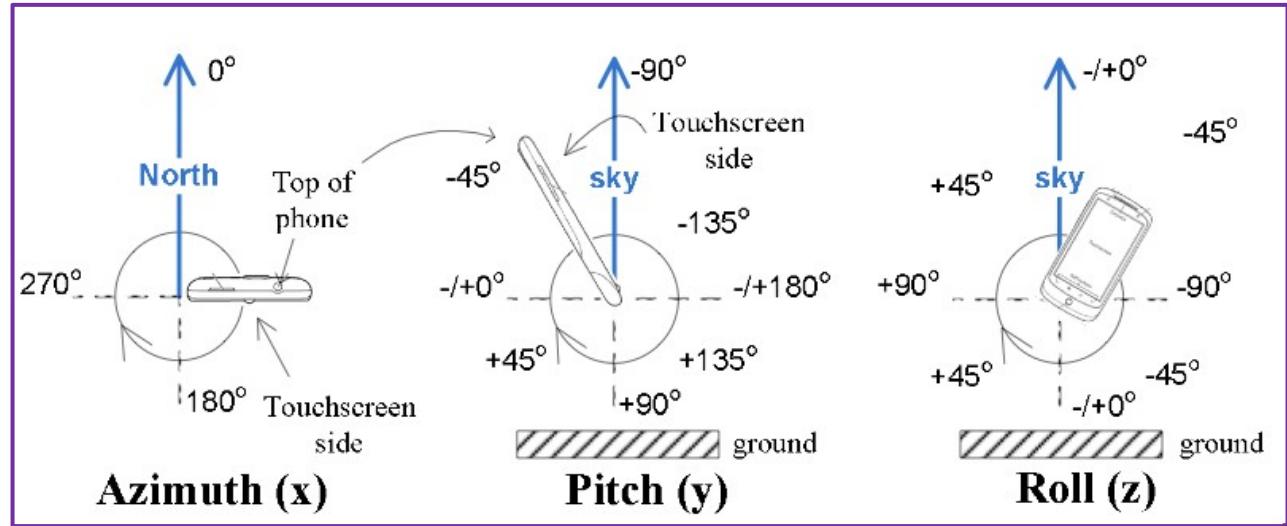
Rowhammer Attack: flip memory bits!

# Property-12: Security

**Cold-boot attack:** physically attack that can scan memory to get any plaintext



# Property-12: Security



**Side-Channel: infer PIN code through motion-detector**

# Property-13: Privacy



ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE STORE

BIZ & IT —

## Private browsing: it's not so private

The private browsing features found in most Web browsers, designed to keep ...

PETER BRIGHT - 8/9/2010, 5:00 AM

84

Research by Stanford University to investigate the privacy of the "private browsing" feature of many Web browsers suggests that the tools aren't all that private after all, and that many kinds of information can be leaked by browsers when using the mode. The paper is due to be presented next week at the USENIX security conference.

f

Twitter icon

"InPrivate Browsing" in Internet Explorer, "Incognito mode" in Chrome, and "Private Browsing" in Firefox and Safari all strive to do the same two things: make it impossible for users of the same computer to figure out which sites the browser has been used to visit, and make it impossible for sites to know whether or not a particular user has previously visited them.

# Property-14: Trust



Fighting Club, 1999

# Conflict between these Properties

Usability VS. Privacy

Performance VS. Security

Performance VS. Energy

Fault Tolerance VS. Consistency

...

All these lead to more complexity



# **SYSTEM COMPLEXITY**

# Problem Types

## Emergent properties (surprise!)

- The properties that are not considered at design time

## Propagation of effects (butterfly effect)

- Small change -> big effect

## Incommensurate scaling (growing pains)

- Design for small model may not scale

## Trade-offs (waterbed effect)

- You cannot sell the cow and drink the milk

# 1. Emergent Properties

## Features

- Not evident in the individual components of a system
- But show up when combining those components
- Might also be called surprises
- An unalterable fact of life: some things turn up **only when a system is built**

# 1. Emergent Properties

## The Millennium Bridge

- For pedestrians over the River Thames in London
- Pedestrians synchronize their footsteps when the bridge sways, causing it to sway even more



- It had to be closed after only a few days

# Emergent Property Example: Ethernet

All computers share single cable

Goal is reliable delivery

Listen while sending to detect collisions

- If two nodes sends data at the same time, then both cancel and wait for a random time



# Does Collision Detection Work?



**What if A finishes sending before data from B arrives?**

- 1km at 60% speed of light = 5 ms (microseconds)
- Original Ethernet Spec: 3 Mbit/sec
  - A can send 15 bits before bit 1 arrives at B
  - A must keep sending for  $2 \times 5$  ms (to detect collision when first bit from B arrives)
- Minimum packet size is  $5 \times 2 \times 3 = 30$  bits
- The default header is 5 bytes (40 bits), so **no problem for now**

# 3 Mbit/s -> 10 Mbit/s, What will Happen?

First Ethernet standard: 10 Mbit/s, 2.5 km wire

- Must send for  $2 \times 12.5 \mu\text{seconds} = 250 \text{ bits} @ 10 \text{ Mb/s}$
- Header was 14 bytes
- Needed to pad packets to **at least 250 bits** (~32 bytes)

Emergent property: **Minimum packet size!**

- The 250-bit minimum packet size is a surprise

## 2. Propagation of Effects [Cole'69]

WHO: tried control malaria in North Borneo

- Sprayed villages with DDT
- Wiped out mosquitoes, but ....
- Roaches collected DDT in tissue
- Lizards ate roaches and became slower
- Easy target for cats
- Cats didn't deal with DDT well and died
- Forest rats moved into villages
- Rats carried the bacillus for the plague

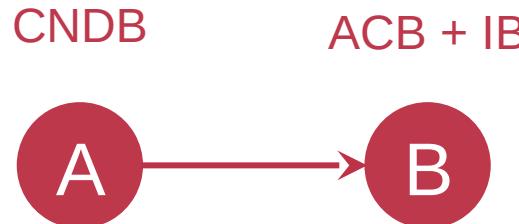
WHO just replaced **malaria** with the **plague**



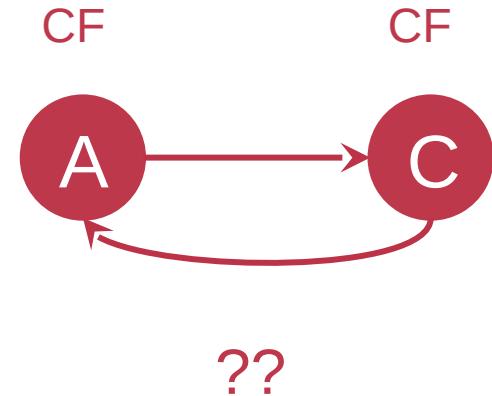
# Example: No Small Changes

## Phone network features

- CF: Call Forwarding
- CNDB: Call Number Delivery Blocking
  - The caller's number should be hidden
- ACB: Automatic Call Back
- IB: Itemized Billing



- A calls B, B is busy
- Once B is done, B automatically calls A
- A's (caller) number appears on B's bill



### 3. Incommensurate Scaling

**As a system increases in size or speed, not all parts of it follow the same scaling rules**

- So things stop working

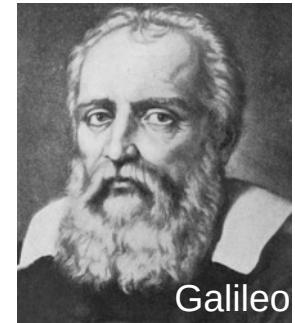
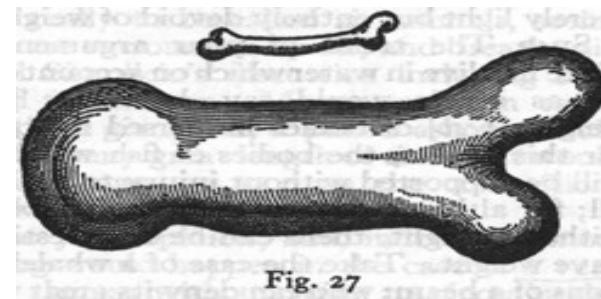
**The mathematical description**

- Different parts of the system exhibit different orders of growth

### 3. Incommensurate Scaling

#### Galileo in 1638

- To illustrate briefly, I have sketched a bone whose natural length has been increased three times and whose thickness has been multiplied until, for a correspondingly large animal, it would perform the same function which the small bone performs for its small animal.
- From the figures here shown you can see how out of proportion the enlarged bone appears.

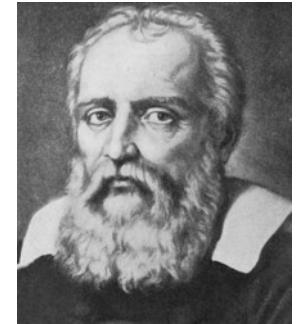


Galileo

### 3. Incommensurate Scaling

#### Galileo in 1638

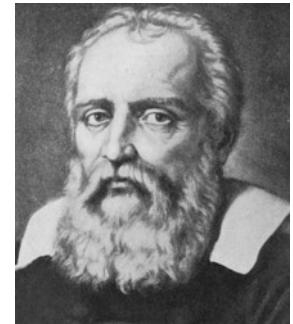
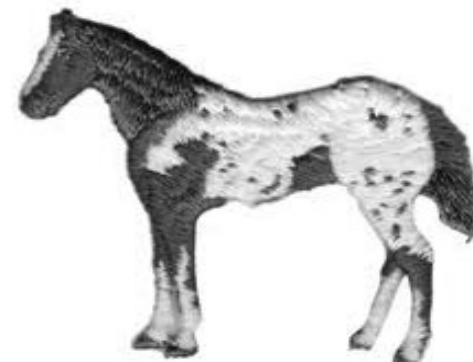
- Clearly then if one wishes to maintain in a great giant the same proportion of limb as that found in an ordinary man he must either find a harder and stronger material for making the bones, or he must admit a diminution of strength in comparison with men of medium stature; for if his height be increased inordinately he will fall and be crushed under his own weight.



### 3. Incommensurate Scaling

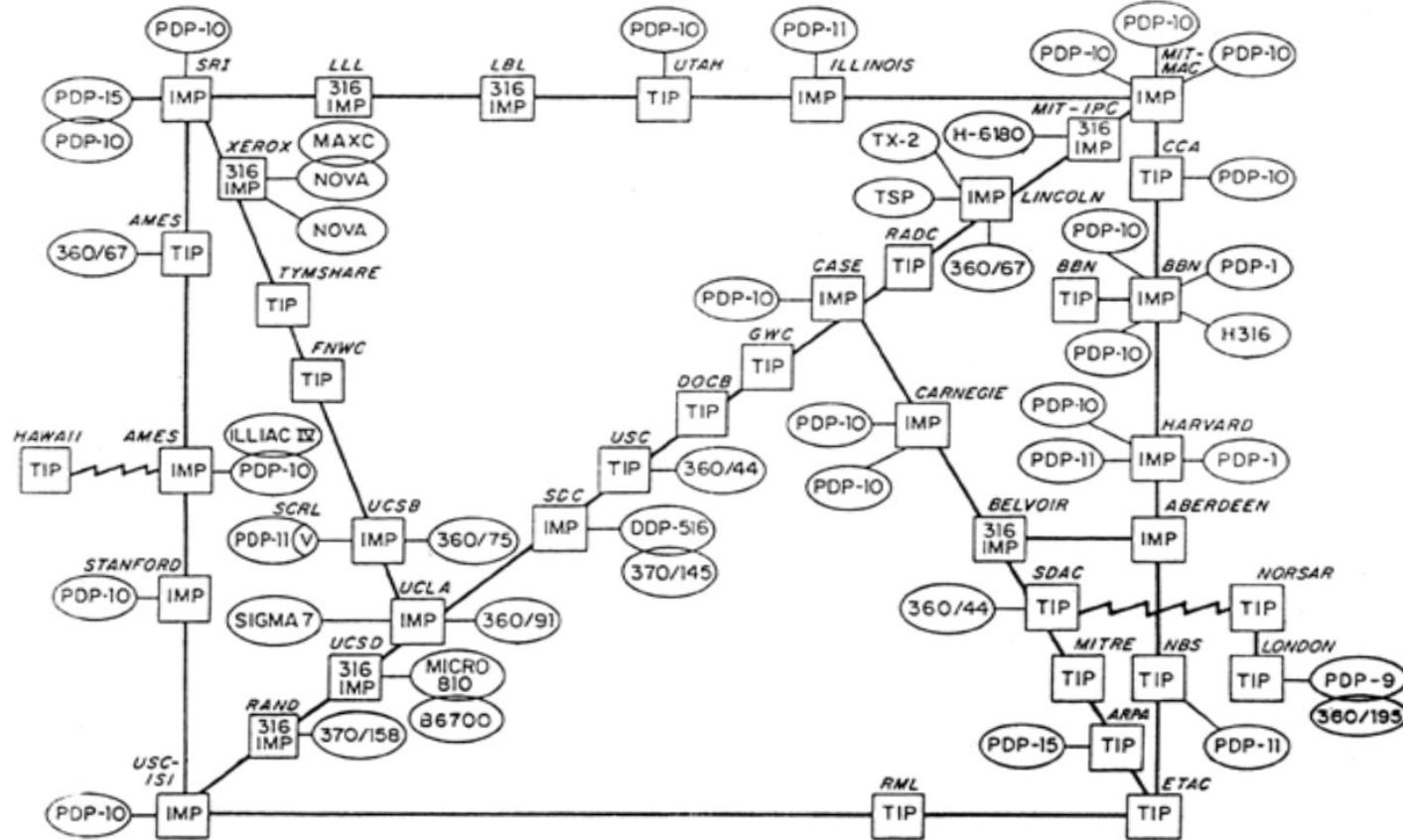
#### Galileo in 1638

- Whereas, if the size of a body be diminished, the strength of that body is not diminished in the same proportion; indeed the smaller the body the greater its relative strength. Thus a small dog could probably carry on his back two or three dogs of his own size; but I believe that a horse could not carry even one of his own size.

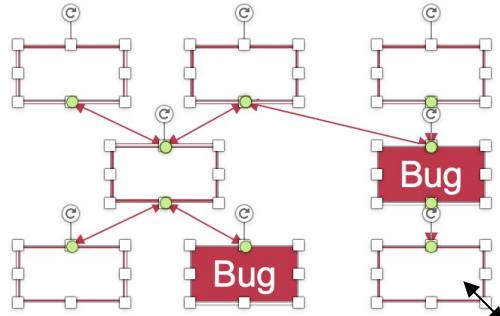


### 3. Incommensurate Scaling

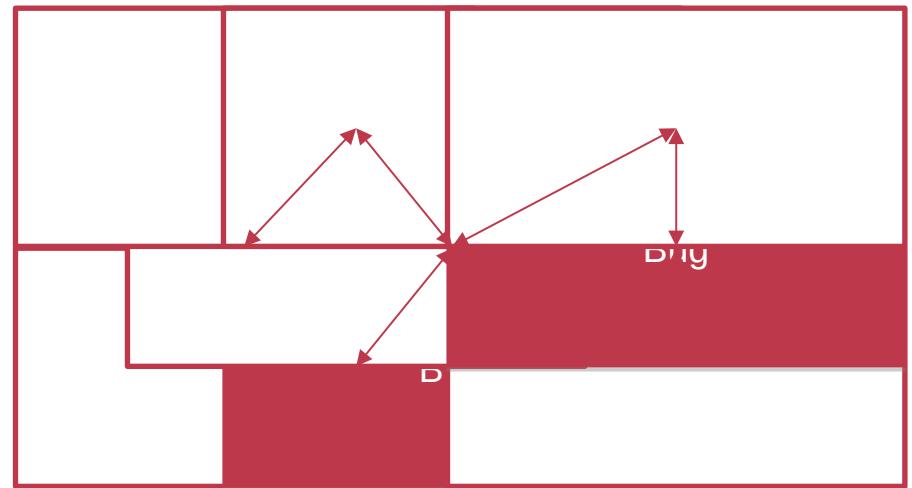
ARPA NETWORK, LOGICAL MAP, SEPTEMBER 1973



# Example: Scaling Figures in PPT



Original



A mess after scaling

# Example: Scaling the Internet

Size routing tables (for shortest paths):  $O(n^2)$

- Hierarchical routing on network numbers
- Address: 16-bit network number and 16-bit host number

Limited networks ( $2^{16}$ )

Solutions:

- NAT (Network Address Translators) and IPv6

## 4. Trade-offs

### General models

- Limited amount of goodness
- Maximize the goodness
- Avoid wasting
- Allocate where helps most

### Waterbed effect

- Pushing down on a problem at one point
- Causes another problem to pop up somewhere else

## 4. Trade-offs

### Binary classification

- We wish to classify a set of things into two categories
  - Based on presence or absence of some property
- But we lack a direct measure of that property
- So we identify some indirect measure instead
  - Known as a proxy

## 4. Trade-offs

### Binary classification (Cont.)

- Occasionally this scheme misclassifies something
- By adjusting parameters of the proxy
- The designer may be able to:
  - reduce one class of mistakes
  - but only at the cost of increasing some other class of mistakes



M.A.L.H

# COPING WITH COMPLEXITY

# How to Handle Complexity?

## Ideally, the "Constructive Theory"

- Allows the designer systematically to
  - Synthesize a system from its **specifications**
  - Make necessary trade-offs with **precision**
- Works in some fields
  - Communication systems
  - Linear control systems
  - Design of bridge and skyscrapers (to a certain extent)



# How to Handle?

## In Computer Systems

- “We find that we were born too soon”
  - In “Principles of Computer System Design: An Introduction”
- The problems
  - We work almost entirely by analyzing ad hoc examples rather than by synthesizing
  - So, in place of a well-organized theory, we use case studies

# Limit the Levels of Complexity

All systems are **indefinitely**

- The deeper one digs, the more signs of complexity turn up
- A computer -> gates -> electrons -> quarks -> ...

Abstraction: limits the depth of digging

# M.A.L.H

## Modularity

- Split up system
- Consider separately

## Abstraction

- Interface/Hiding
- Avoid propagation of effects

## Layering

- Gradually build up capabilities

## Hierarchy

- Reduce connections
- Divide-and-conquer

# Modularity

Analyze or design the system as a collection of interacting subsystems

- Subsystems called **modules**
- "Divide-and-conquer" technique

Modularity is the **simplest, most important tool** for reducing complexity

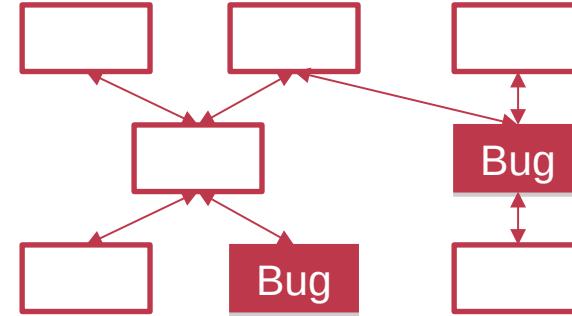
- Consider interactions among the components within a module without simultaneously thinking about the components that are inside other modules

# Modularity



$BugCount \sim N$   
 $DebugTime \sim N \times BugCount$   
 $\sim N^2$

Original System



$$\begin{aligned} DebugTime &\sim \left(\frac{N}{K}\right)^2 \times K \\ &\sim \frac{N^2}{K} \end{aligned}$$

System with Modularity

Assumptions: consider the number of bugs is proportional to its size, and bugs are randomly distributed

# Abstraction

## Abstraction

- Treat a module based on external specifications, no need for details inside

## Principles to divide a module

- Follow natural or effective boundaries
- Fewer interactions among modules (Chap.4 & 5)
- Less propagation of effects

# Abstraction

**Problem: minimizing interconnections among modules may be defeated**

- Unintentional or accidental interconnections, arising from implementation errors
- Well-meaning design attempts to sneak past modular boundaries
  - Improve performance
  - Meet some other requirement

# Abstraction

## Software is particularly subject to this problem

- The modular boundaries provided by the separately compiled subprograms are actually somewhat **soft**
- Is easily penetrated by errors in:
  - using pointer;
  - filling buffer;
  - calculating array index;
  - etc.

# Abstraction

## Failure containment

- When a module does not meet its abstract interface specifications
- Limiting the impact of faults
  - Well-designed and properly enforced modular abstractions
  - Control propagation of effects
- Modules are the units of fault containment

# Layering

**Goal: Reduce module interconnections even more**

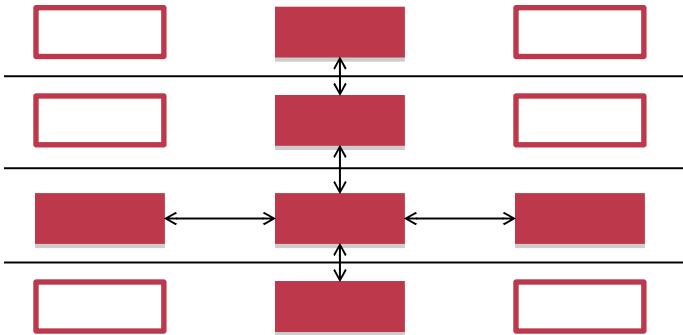
**How to do it?**

- Build a set of mechanisms first (a lower layer)
- Use them to create a different complete set of mechanisms (an upper layer)

**General rule: A module in one layer only interacts with:**

- its peers in the same layer, and
- modules in the next lower layer / next higher layer

# Layering



Application layer

OS layer

Processor & memory layer

Memory cells & gates layer

## House:

- Inner layer of studs, joist, rafter (shape & strength )
- Layer of sheathing and drywall (wind out)
- Layer of siding, flooring and roof tiles (watertight)
- Cosmetic layer of paint (looks good)

## Algebra:

- integer, complex number, polynomials, polynomials with polynomial coefficients

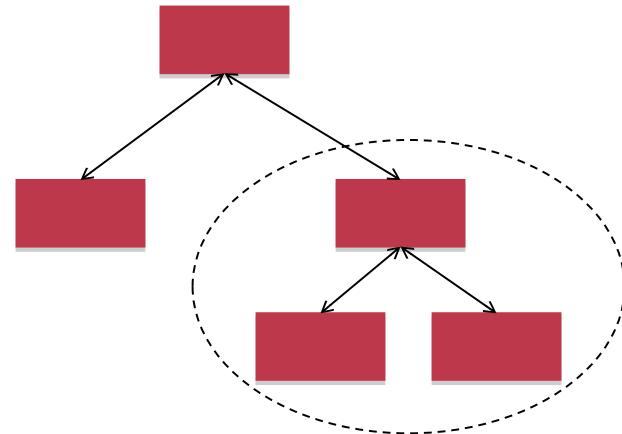
# Hierarchy

## Hierarchy: another module organization

- Start with a small group of modules
- Assemble them into a stable, self-contained subsystem with well defined interface
- Assemble a group of subsystems to a larger subsystem

### Example

- 1 manager leads N employees
- 1 higher manager leads N lower managers



# Hierarchy

**There are many striking examples of hierarchy**

- E.g., microscopic biological systems, assembly of Alexander's empire
- Under evolution, such designs have a better chance of survival

**Constrains interactions**

- Permit interaction only among the components of a subsystem
- Reduces the number of potential interactions among modules from **square-law** to **linear**



**COMPUTER SYSTEMS ARE DIFFERENT**

# Computer Systems are Different

**Computer systems are the same as all other systems**

- Certain common problems show up in all complex systems
- The techniques that have been devised for coping with complexity are universal

**Yet, computer systems are **different** (sadly)**

- The complexity is not limited by physical laws
- The rate of change of technology is unprecedented

# Unbounded Composition

## Two properties of computer systems

- 1. Mostly digital
- 2. Controlled by software
- Both relax the limits on complexity arising from physical laws in other systems

# Computer System: Coping with Complexity

M.A.L.H are NOT enough

- Hard to choose **the right** modularity
- Hard to choose **the right** abstraction
- Hard to choose **the right** layer
- Hard to choose **the right** hierarchy



## COURSE INFO

# Scores

Mid-Exam: 25%



Final-Exam: 35%



Lab: 30%



Homework: 10%



# Where is CSE in Courses

Operating  
Systems

Architecture

Compilers

Networking

Computer Systems Engineering

Introduction to Computer Systems

# Course Info

## Instructors

- Yubin Xia: [xiayubin@sjtu.edu.cn](mailto:xiayubin@sjtu.edu.cn)
- Xingda Wei: [wxdwfc@sjtu.edu.cn](mailto:wxdwfc@sjtu.edu.cn)

## TAs

- Le Chen
- Ke Cheng
- Jingkai He
- Yuhan Yang

## Website

- <https://oc.sjtu.edu.cn/courses/60116>