# What Next?
# A Few Remaining Problems in Information Technology

Jim Gray,

1998 Turing Lecture

ACM FCRC Atlanta

Gray@Microsoft.com

http://Research.Microsoft.com/~Gray

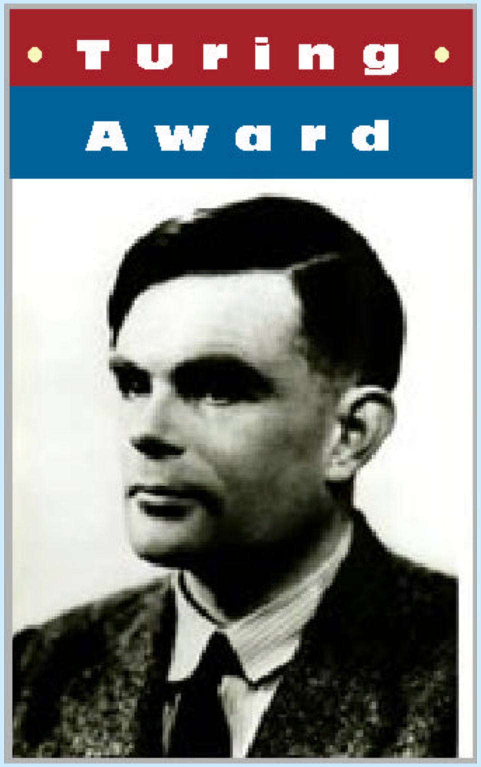
# Outline

- **The need for long-range research and the need for Universities to do some of it. and the need for government support.**

- **Some long-range systems research goals.**

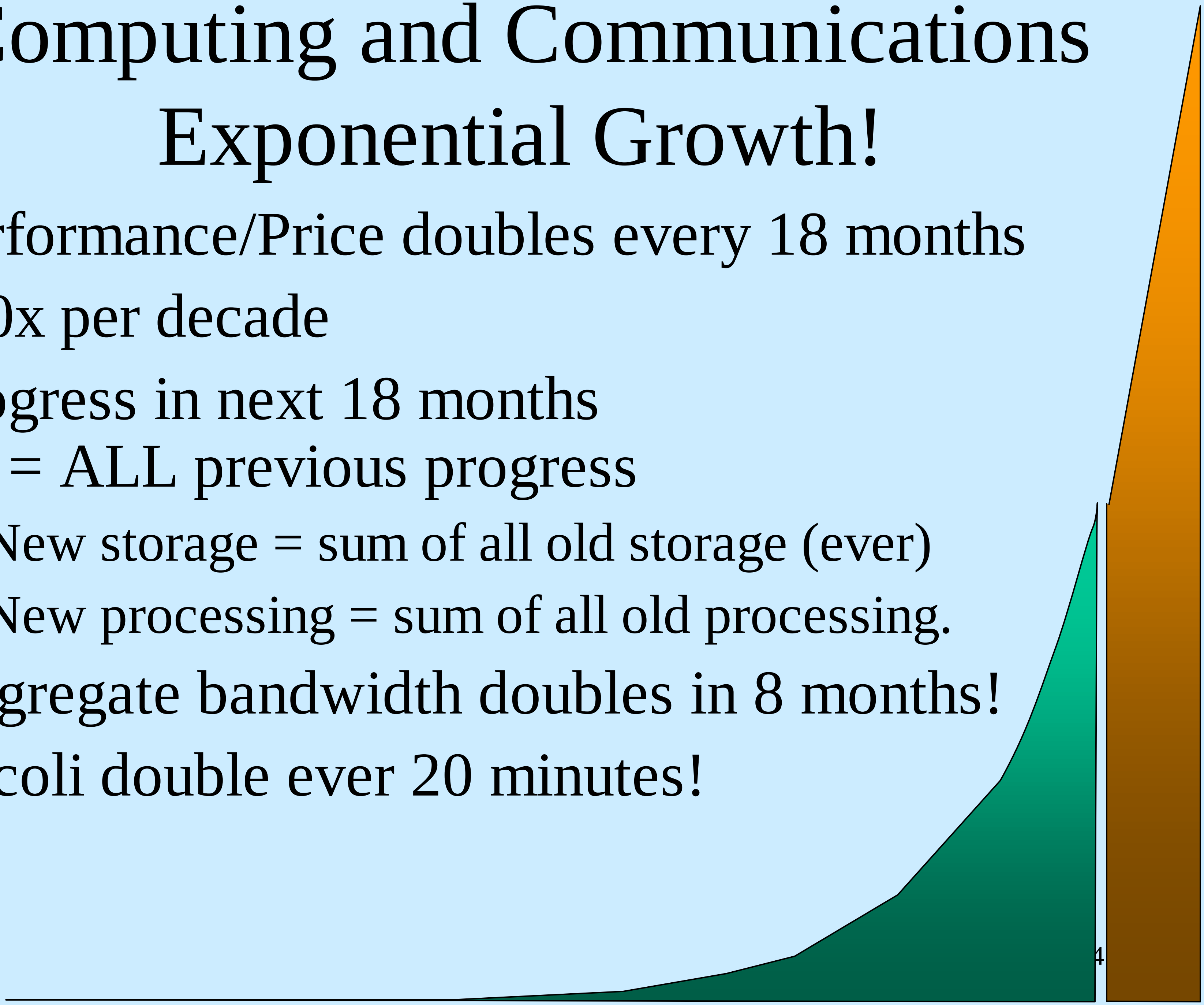
# Thank you!

- ACM Awards Committee
- Lucent.
- Colleagues in
  - Databases,
  - Transaction Processing, and
  - Operating systems.
- We all did this together!



# Computing and Communications Exponential Growth!

- Performance/Price doubles every 18 months
- 100x per decade
- Progress in next 18 months
  = ALL previous progress
  – New storage = sum of all old storage (ever)
  – New processing = sum of all old processing.
- Aggregate bandwidth doubles in 8 months!
- E. coli double ever 20 minutes!



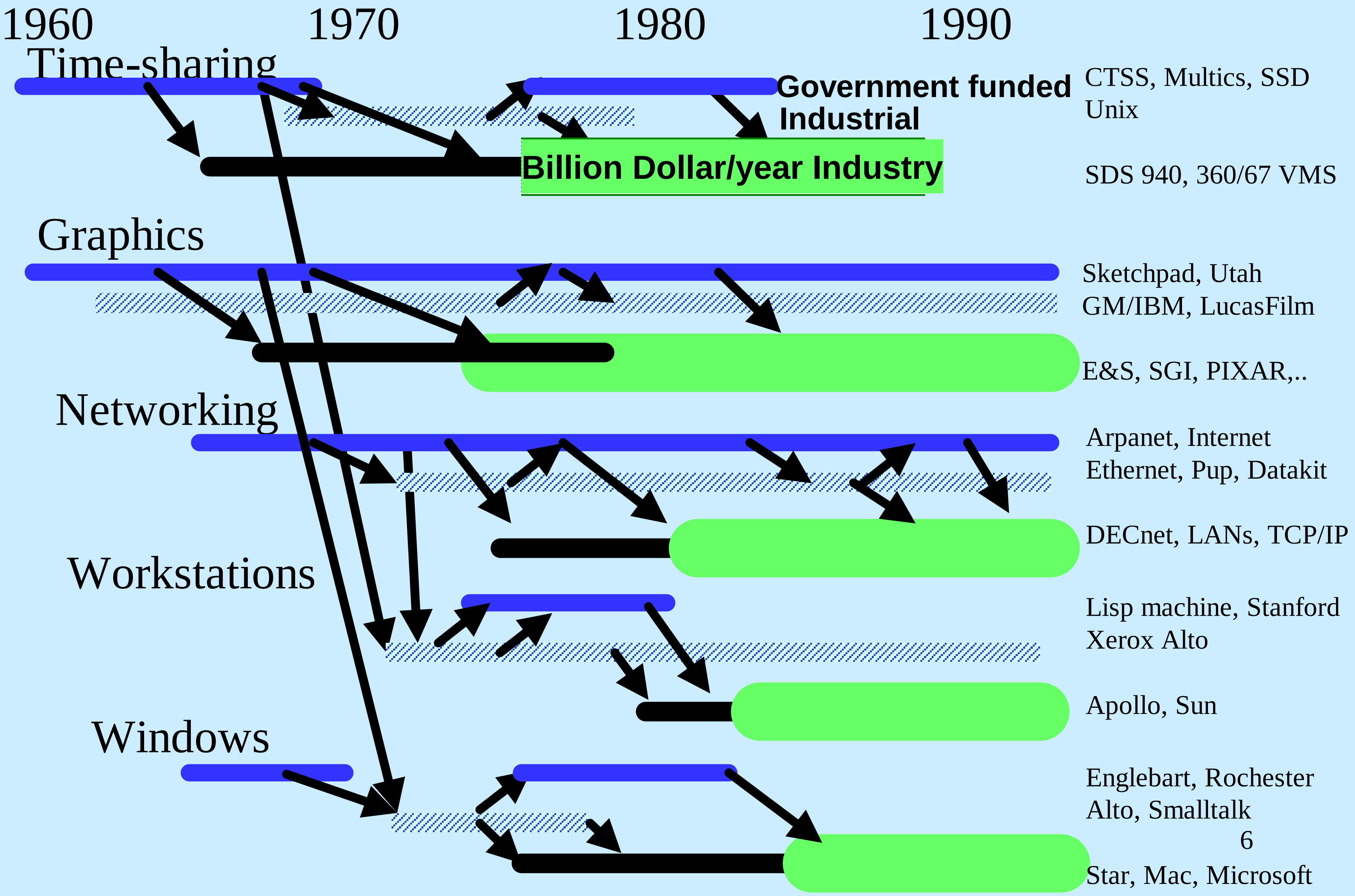
# Cyberspace is a *New World*.

- We have discovered a “new continent”.
- It is changing how we learn, work, and play.
  - 1 T$/y industry
  - 1 T$ new wealth since 1993
  - 30% of US economic growth since 1993
- There is a gold rush to stake out territory.
- But we also need explorers:
  Lewis & Clark expeditions
  Universities to teach the next generation(s)
- Governments, industry, and philanthropists
  should fund long-term research.

THE
LONG
BOOM

# Research Investments Pay Off

CSTB –NRC Evolving the High-Performance Computing and Communications Imitative to Support the nations Information Infrastructure, NA Press, Washington DC, 1995.



1960 1970 1980 1990

Time-sharing

**Government funded**

**Industrial**

**Billion Dollar/year Industry**

CTSS, Multics, SSD
Unix

SDS 940, 360/67 VMS

Graphics

Sketchpad, Utah
GM/IBM, LucasFilm

E&S, SGI, PIXAR,..

Networking

Arpanet, Internet
Ethernet, Pup, Datakit

DECnet, LANs, TCP/IP

Workstations

Lisp machine, Stanford
Xerox Alto

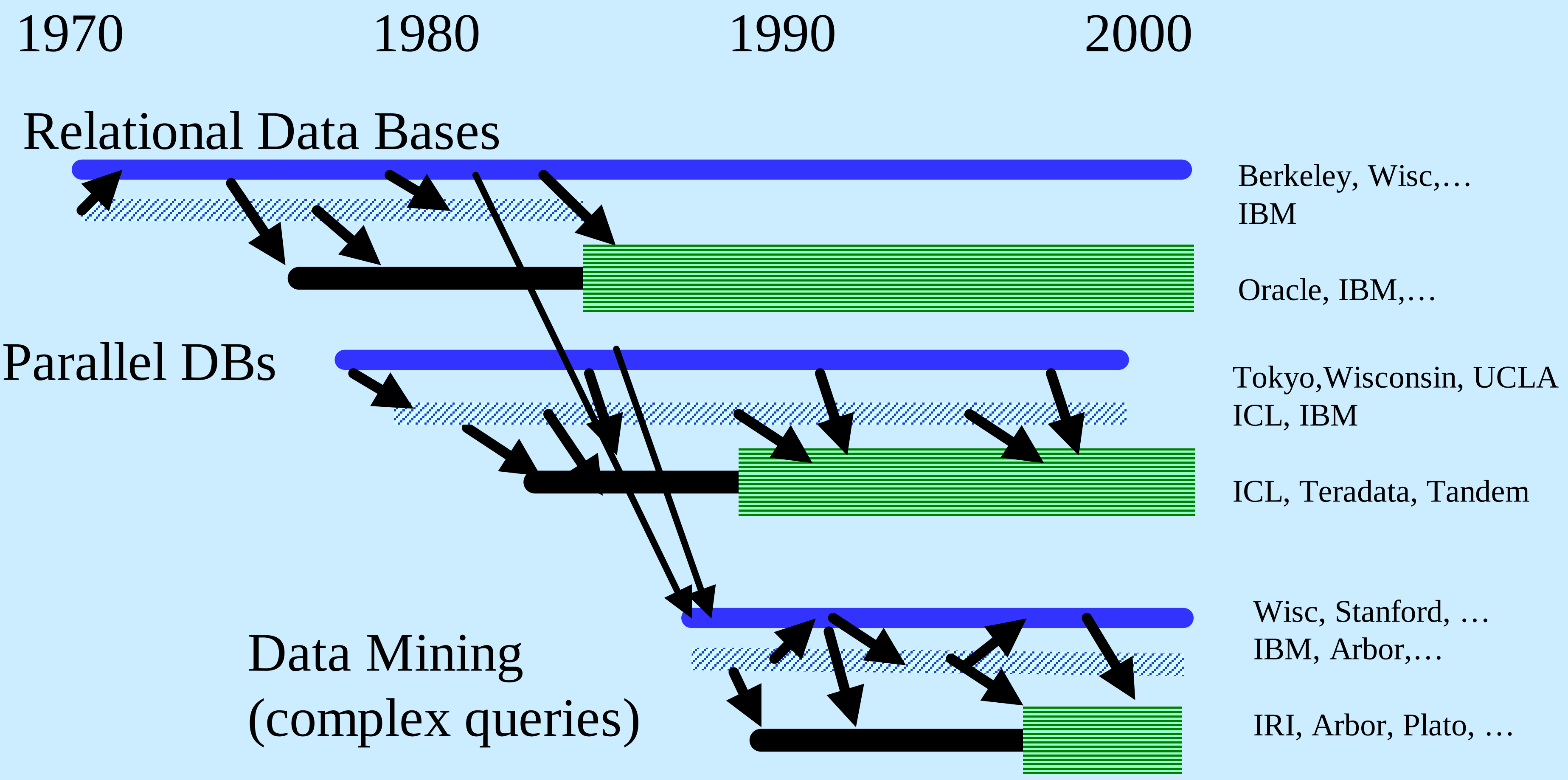
Apollo, Sun

Windows

Englebart, Rochester
Alto, Smalltalk

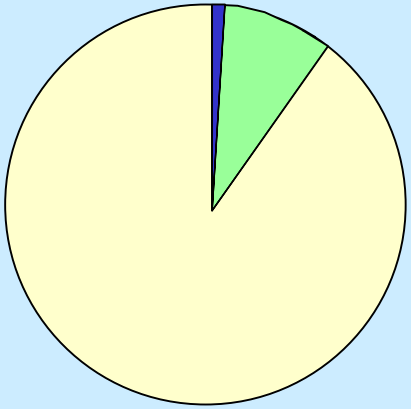
Star, Mac, Microsoft

# Research Investments Pay Off



# Why Can’t Industry Fund IT Research?

- **It does**: IBM (5.8%), Intel(13%), Lucent (12%), Microsoft(14.%), Sun (12%), ...
  - R&D is ~5%-15% (50 B$ of 500 B$)
    - AD is 10% of that (5 B$)
      - Long-Range Research is 10% of that 500 M$ 2,500 researchers and university support



  - Compaq: 4.8% R&D (1.3 B$ of 27.3 B$).AOL: 3.7% D, ?R (96 M$ of 2.6 B$)
  - Dell:1.6% R&D (204 M$ of 12.6 B$), EDS, MCI-WorldCom, ….
- To be competitive, some companies cannot make large long-term research investments.

The Xerox/PARC story:

created Mac, Adobe, 3Com…

# PITAC Report

Presidential IT Advisory Committee

http://www.ccic.gov/ac/report/

- Findings:
  - – Software construction is a mess: needs breakthroughs.
  - – We do not know how to scale the Internet 100x
    - Security, manageability, services, terabit per second issues.
  - – USG needs high-performance computing (Simulation) but market is not providing vector-supers – just providing processor arrays.
  - – Trained people are in very short supply.
- Recommendations:
  - – Lewis & Clark expeditions to 21st century.
  - – Increase long-term research funding by 1.4B$/y.
  - – Re-invigorate university research & teaching.
  - – Facilitate immigration of technical experts.

# Outline

- **The need for long-range research and
the need for Universities to do some of it.
and the need for government support.**

- **Some long-range systems research goals.**

# Properties of a Research Goal

- Simple to state.
- Not obvious how to do it.
- Clear benefit.
- Progress and solution is testable.
- Can be broken in to smaller steps
    - So that you can see intermediate progress.

# I was motivated by a simple goal

1. ***Devise an architecture that scales up: Grow the system without limits*[*]*.***

scaleup:
1,000,000 : 1

This is impossible *(without limits?)*, but...

This meant automatic parallelism,
automatic management,
distributed,
fault tolerant,
high performance

- Benefits:
  - long term vision guides research problems
  - simple to state, so attracts colleagues and support
  - Can tell your friends & family what it is that you do ☺.

# What I did

- Transaction model (and implementation).
- Throughput metric (transaction per second).
- High-availability systems.
- Parallel Database systems.
- Data cube operator.

# Transaction Model

- State consists of entities with values.
- Execution consists entity read/write actions.
- Groups of actions are transactions.
- How to avoid concurrency anomalies:
  - Def of anomalies: not equivalent to serial execution
  - All transactions are well-formed and 2 phase
    iff no concurrency anomalies (almost).
- ACID properties:
  - Atomicity, Consistency, Isolation, Durability
- Many techniques: locks, logs, ....
- Two-Phase commit for distributed system atomic commit.

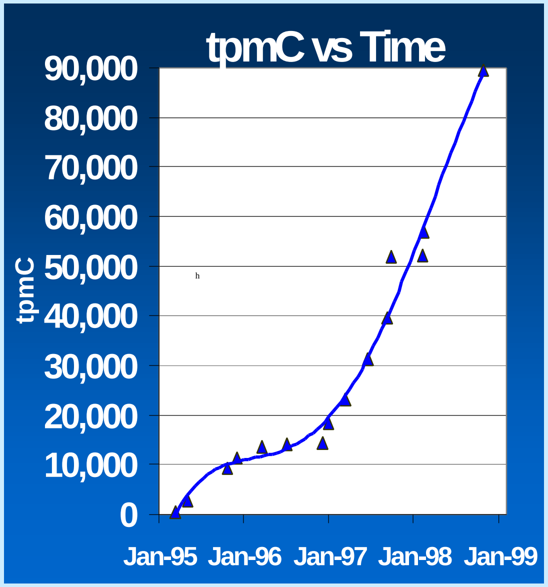
# Performance

- How to measure it:
  - Transactions per second
  - Needed a standard transaction (highly portable)
  - Included a standard price/performance metric.
- Standard Performance and Price/Performance metrics drove great progress.
- Benchmarking spreads best practice.
- 1 B transactions per day

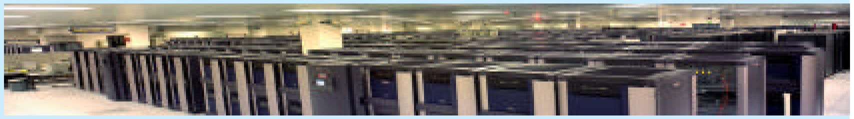
# Highly Available Systems

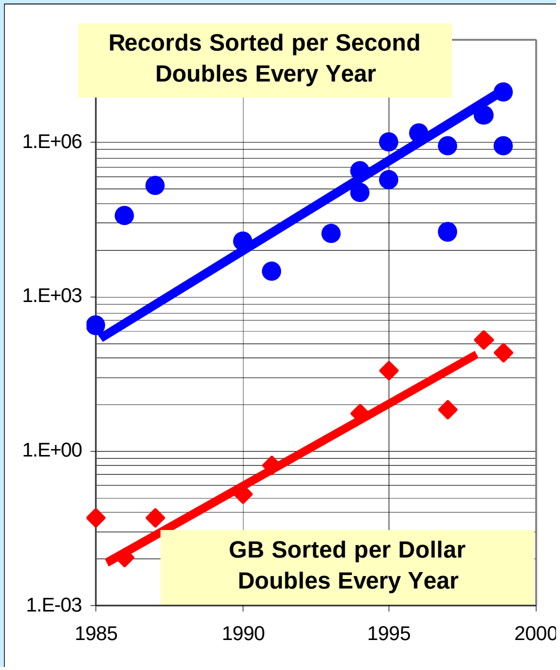
- Did end-to-end measurements of system’s availability.
- Showed that in the end, it is all software/design faults.
- Worked on software-fault tolerance.

# Scalability Successes



- Internet
  - 100 M “nodes”
  - But PITAC angst about the future.
- Single Site Clusters
  - Billions of transactions per day
  - Tera-Ops & Peta-Bytes (10 k node clusters)
  - Micro-dollar/transaction
- Hardware + Software advances
  - TPC & Sort examples (2x/year)
  - *Many* other examples





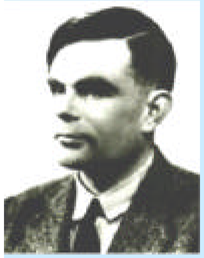
# Scalability Embarrassments

- Still…
  - We have yet to make parallel programming easy
  - Only automatic parallelism has “won”
    - Parallel decision support databases
    - Parallel oltp/web/file/print/…
  - Managing computer clusters is a major cost.
- New computer architectures will be highly parallel
    - 100 instruction steams per chip in a decade.
    - 10$ MEMS **MicroElectroMechanical Systems** million-node systems
- So, the ScaleUp problem is not solved.

# Three Seminal Papers

- Babbage: Computers
- Bush: Automatic Information storage & access
- Turing: Intelligent Machines

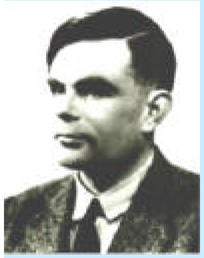
- Note:
  - Previous Turing lectures described several “theory” problems.
  - Problems here are “systems” problems.
  - Some include a “and prove it” clause.
  - They are enabling technologies, not applications.
  - Newell’s: Intelligent Universe (Ubiquitous computing.) missing because I could not find “simple-to-state” problems.



# Alan M.Turing (1912-1954)

Computing machinery and intelligence. *Mind*, Vol. LIX. 433-460, 1950

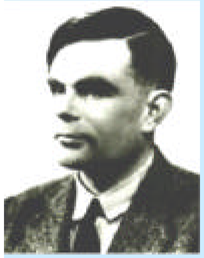
- Computers will be intelligent.
- Debate then and now:
  - Will this just be a symbiotic relationship (computer as tool)?
  - Or will computers be “conscious”?



# The Turing Test

- Imitation Game:
  - Judge, man, and a woman
  - All chat via Email.
  - Man pretends to be a woman.
  - Man lies, woman tries to help judge.
  - Judge must identify man after 5 minutes.

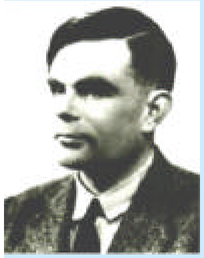
2. Turing Test
   - Replace man or woman with a computer.
   - Fool judge 30% of the time.



# What Turing Said

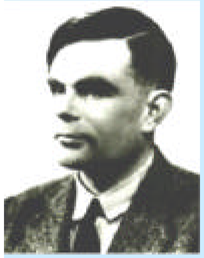
“I believe that in about fifty years' time it will be possible, to programme computers, with a storage capacity of about $10^9$, to make them play the imitation game so well that an average interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning. The original question, "Can machines think?" I believe to be too meaningless to deserve discussion. Nevertheless I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted.”

Alan M.Turing, 1950<br>
“Computing machinery and intelligence.” *Mind*, Vol. LIX. 433-460



# 49 Years Later

- Turing’s technology forecast was great!
  - Billion byte memory is common.
- Intelligence forecast was optimistic.
  - Several internet sites offer Turning Test chatterbots.
  - None pass (yet) http://www.loebner.net/Prizef/loebner-prize.html
- But I believe it will not be long
  (less than 50 years, more than 10 years).
- Turing test still stands as a long-term challenge.



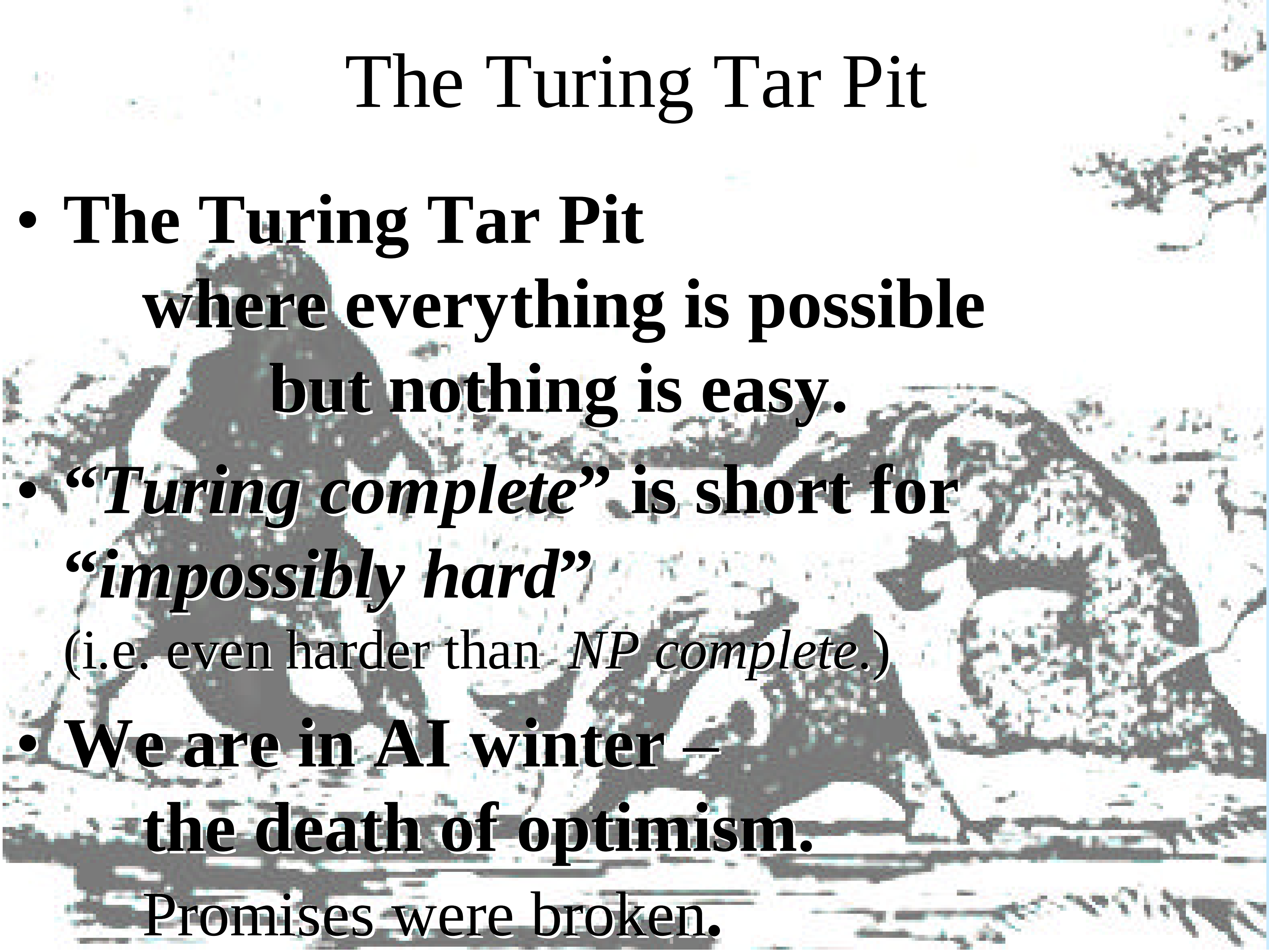
# There Has Been Progress

- Computers helped with
  - endgame proof of the 4-color problem.

    K. Appel and W. Haken, "The solution of the four-color-map problem," *Scientific American*, Oct 1977, 108-121
    and for a "manual" proof: http://www.math.gatech.edu/~thomas/FC/fourcolor.html (1995)
- Computer beat world chess champion
  - with some help from its programming staff (!)
- Computers help design most things today.

- These are Symbiotic Relationships
- **Learning and Concept formation are still an elusive goal.**

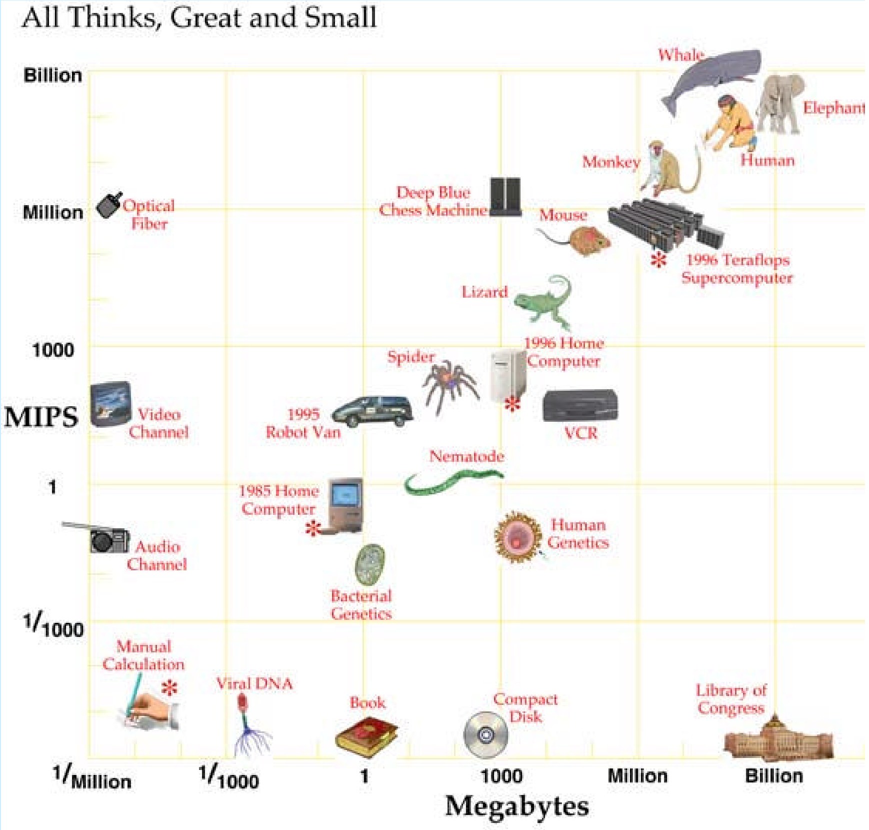
# The Turing Tar Pit

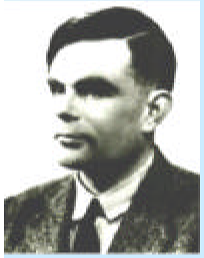


- **The Turing Tar Pit**
  **where everything is possible**
  **but nothing is easy.**
- **“*Turing complete*” is short for “*impossibly hard*”**
  (i.e. even harder than *NP complete*.)
- **We are in AI winter –**
  **the death of optimism.**
  Promises were broken.

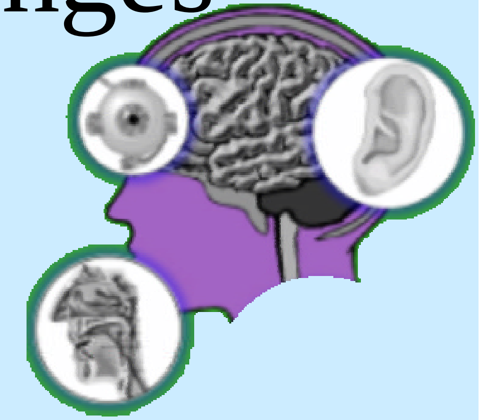
# The Counting Trap

- Humans are 100 tera-bytes of information. and 100 tera-ops
  Figure from ROBOT, Hans Moravec, Oxford, 1998, page 58
- So, a super-computer has comparable power.
- Genome is $10^9$ bits
  - 90% junk
  - 90% common to chimpanzees.
  - 90% common among individuals
  - So really only $10^6$ bytes (huh?!)
- We are missing something:
  - Great compression algorithm?
  - Better programming language?
  - Learning?





# Prosthetics: 3 more challenges



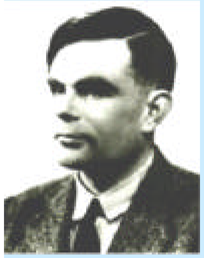
- Implicit in the Turing Test:

  Read and understand as well as a human

  Think and write as well as a human

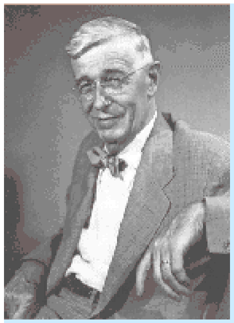
3. Hear as well as a person (native speaker): speech to text
4. Speak as well as a person (native speaker): text to speech
5. See as well as a person (recognize objects and behavior).

- Illustrate as well as a person (done!)
  but virtual reality is still a major challenge.
  create realistic 3D scenes in real time
- Remember what is seen and heard
  and quickly return it on request.



# Benefits of Prosthetics

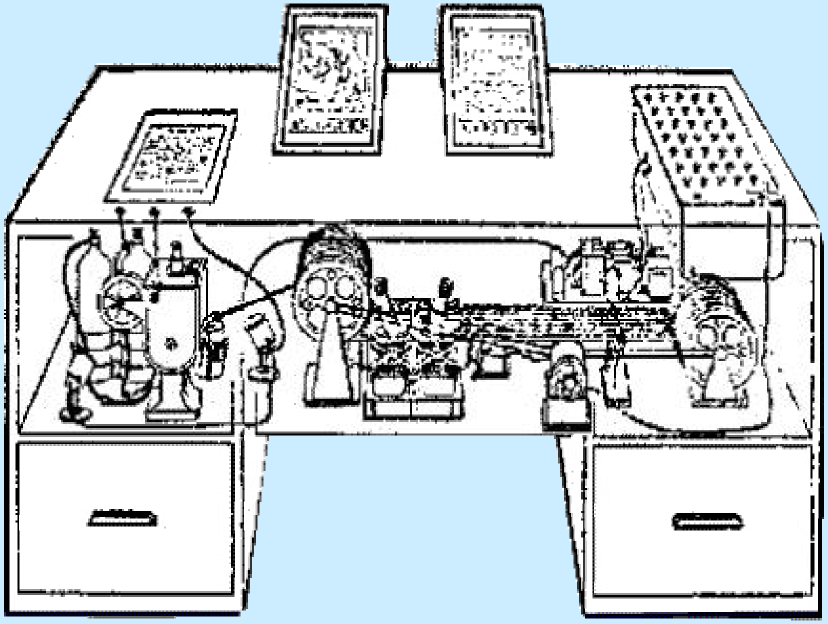
- Today:
  - – computers read for the blind (OCR &text to speech)
  - – Hear for the deaf (speech to text)
  - – Type for the impaired (speech to text).
- Soon:
  - – Prosthetics for all of us (better memory, vision, …)
  - – Communication tools
    - Translating telephones ...
  - – **Revolutionize the human-computer interface.**



# Vannevar Bush (1890-1974)

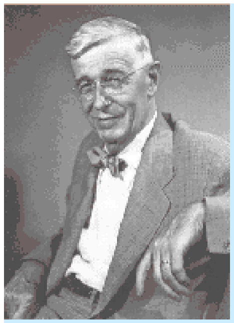
” As We May Think” *The Atlantic Monthly*, **July 1945**

http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm



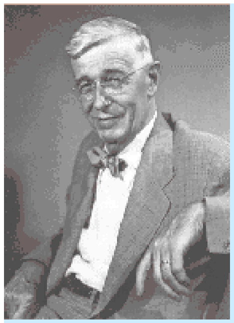
- Memex
  **All human knowledge**
  **in Memex**
  **“a billion books”**
  **hyper-linked together**
- **Record everything you see**
  - **camera glasses**
  - **“a machine which types when talked to”**
- **Navigate by**
  **text search**
  **following links**
  **associations.**

- **Direct electrical path to human nervous system?**



# Memex is Here! (or near)

- The Internet is growing fast.
- Most scientific literature is online somewhere.
  - – it doubles every 10 years!
- Most literature is online (but copyrighted).
- Most Library of Congress visitors: web.
- A problem Bush anticipated:
  Finding answers is hard.



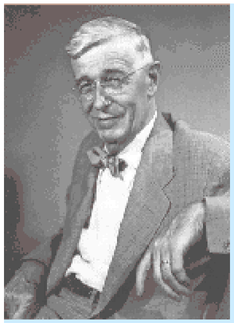
# Why information moves to cyberspace.

- **Low rent: 10x cheaper**

  100 letters on disk: 10¢ in file cabinet 500¢

  1 picture: on disk: 10¢ printed 40 ¢

- **Easy access and search:**
  - Robot can find all docs matching a predicate
  - Access from anywhere
  - Human costs 15$/hr



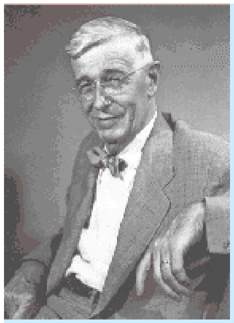
# Why Valuable Information **Not** Available Online

- Owners fear intellectual property will be stolen.
- Most information on the web is paid for by advertising (seems to be free).
- There are copy-protection & payment schemes:

1. Allow owner to be paid for all use (according to contract terms).
2. allows viewers/listeners easy and anonymous viewing.

- Issues are technical, legal, and business.

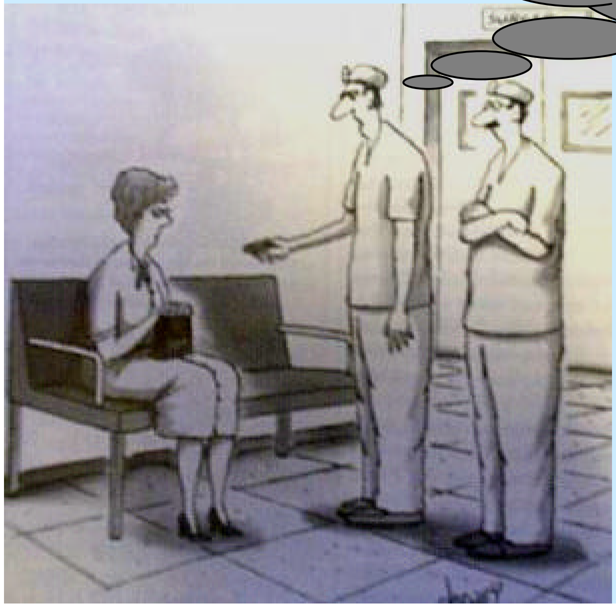
Better schemes will probably be invented.



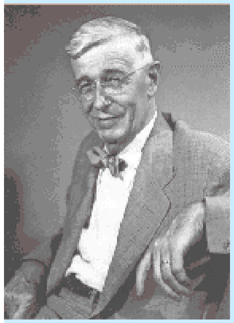
# Personal Memex

6. Remember what is seen and heard
   and quickly return any item on request.

*Your husband died,*
*but here is his black box.*

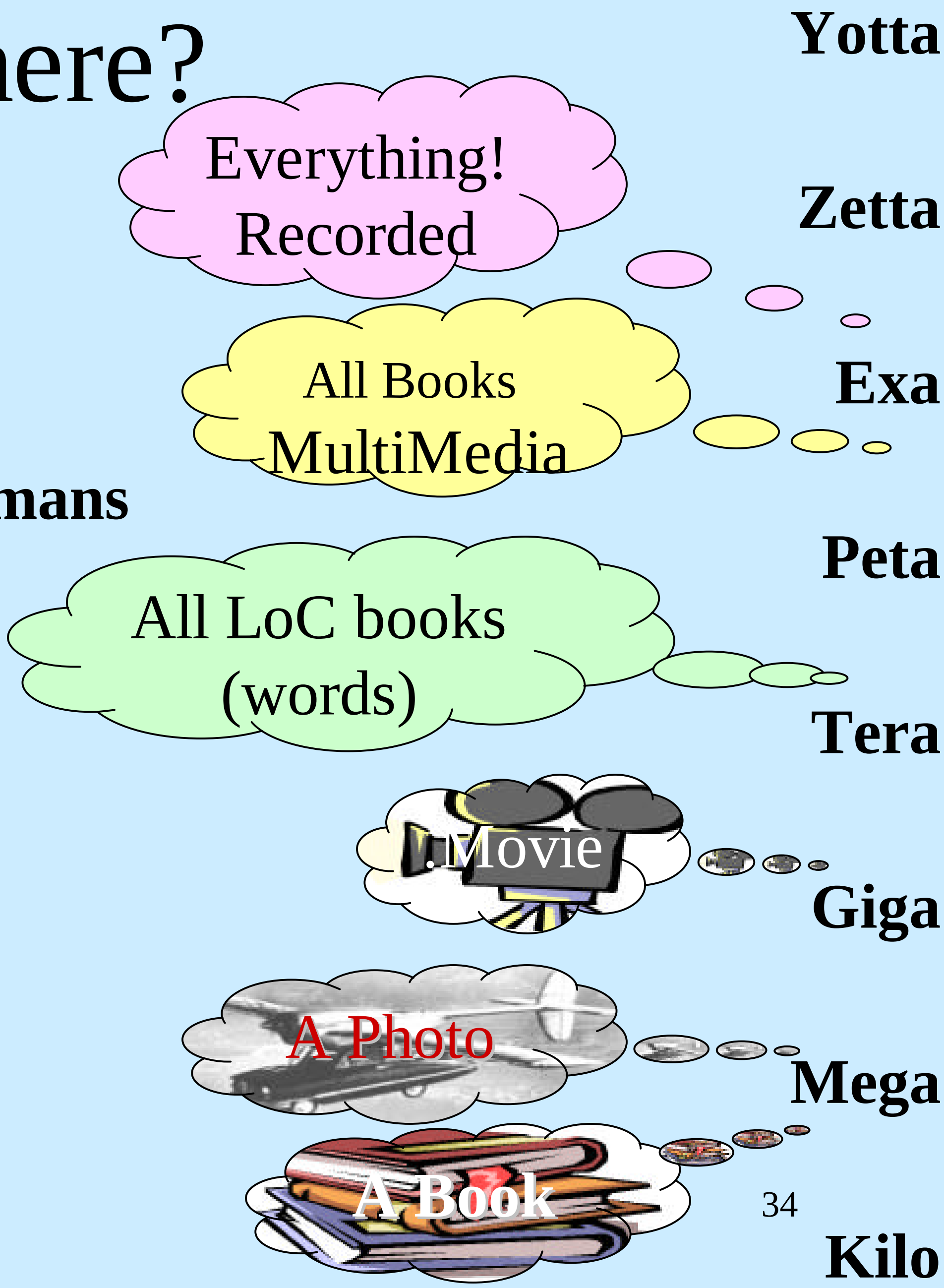


| Human input data | /hr | /lifetime |
|---|---|---|
| **read** text | 100 KB | 25 GB |
| **Hear** speech @ 10KBps | 40 MB | 10 TB |
| **See** TV@ .5 MB/s | 2 GB | 8 PB |



# How Much Information Is there?

- **Soon everything can be recorded and indexed**
- **Most data never be seen by humans**
- **Precious Resource: Human attention**
  Auto-Summarization
  Auto-Search
  is key technology.
  www.lesk.com/mlesk/ksg97/ksg.html



Yotta

Zetta

Exa

Peta

Tera

Giga

Mega

Kilo

Everything! Recorded

All Books MultiMedia

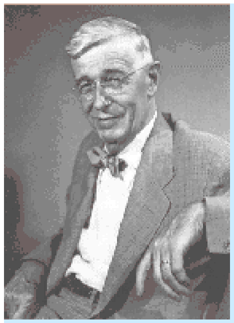
All LoC books (words)

.Movie

A Photo

A Book

24 Yecto, 21 zepto, 18 atto, 15 femto, 12 pico, 9 nano, 6 micro, 3 milli



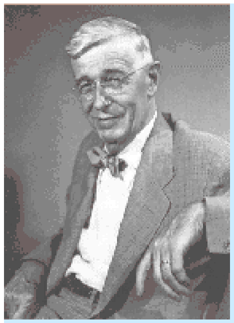
# The Librarian
# Find and Summarize Information

7. Build a system that, given a text corpus, can answer questions about the text and summarize it
   - As precisely as a human expert in that field.
   - As quickly as a human expert in that field.

Do the same for:

- Sounds: conversations, music
- Images: pictures, art, movies,…

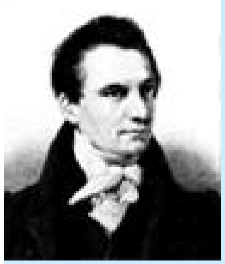
- Note:
  This will be a multi-media interface:
  vision, speech, gestures, graphics,... (not just language)



# TelePresence: Extended Memex

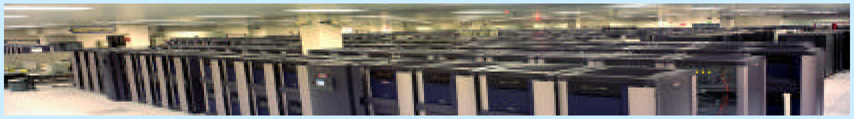
8. Simulate being some other place
   - As an observer (Tele-Observe)
     - Hear& see as well as actually being there.
     - TV gives a low quality and no control  experience.
   - As a participant (Tele-Present)
     - Interact with others as though you are there.
     - Chat & telephone give text & audio tele-presence

- Will happen first in Cyberspace,
  - then perhaps in real space.



# Charles Babbage (1791-1871)

- Babbage's computing goals have been realized
  - But we still need better algorithms & faster machines
- What happens when



  - Computers are free and infinitely powerful?
  - Bandwidth and storage is free and infinite?
- Remaining limits:
  - Content: the core asset of cyberspace
  - Software: Bugs, >100$ per line of code (!)
  - Operations: > 1,000 $/node/year

# ops/s/$ had 3 growth curves: 1890-1950

**1890-1945**

**Mechanical**

**Relay**

**7-year doubling**

**1945-1985**

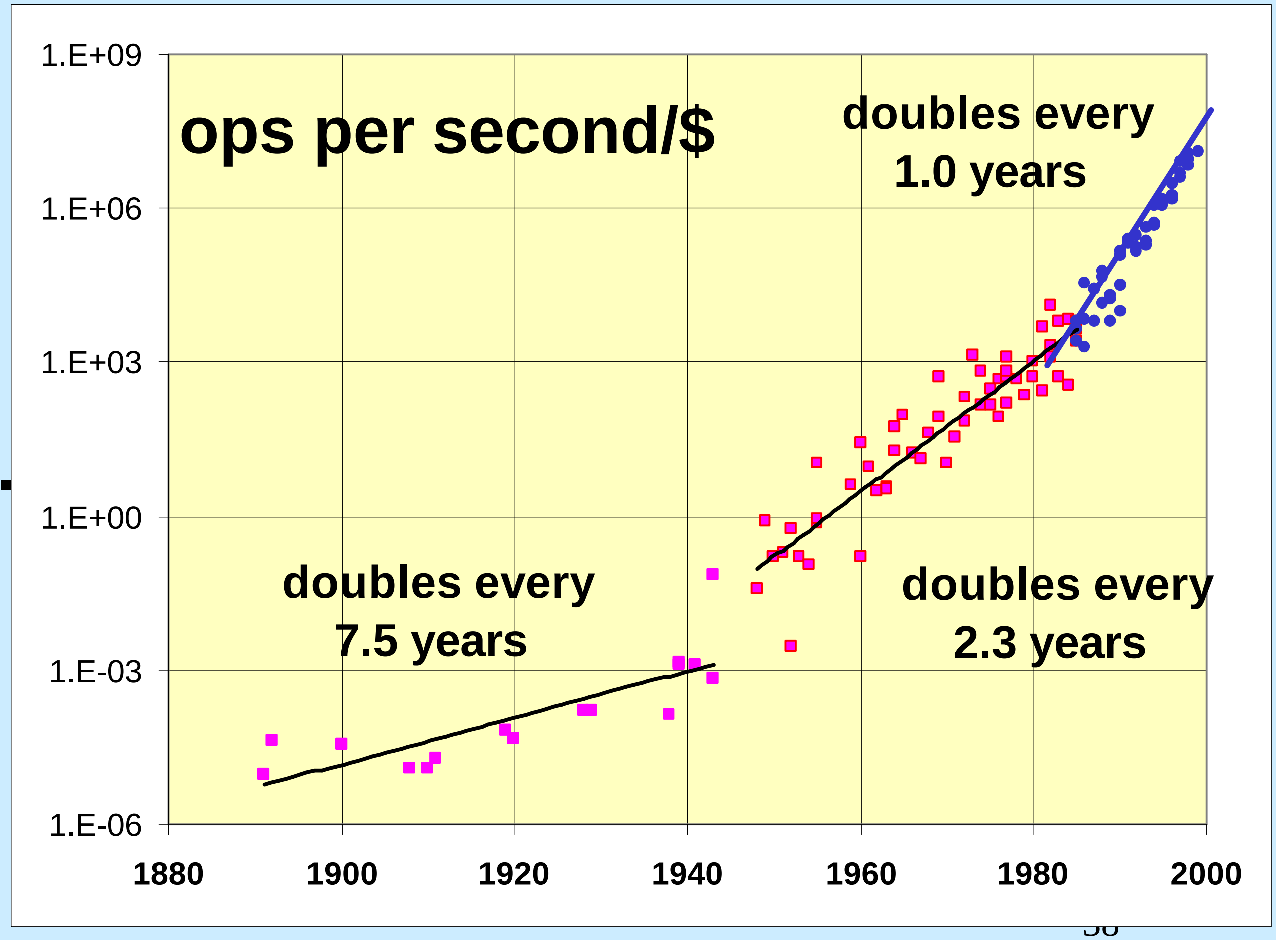
**Tube, transistor,..**

**2.3 year doubling**

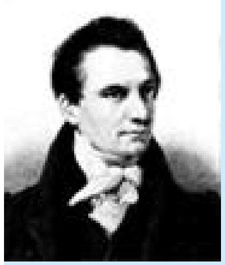
**1985-2000**

**Microprocessor**

**1.0 year doubling**

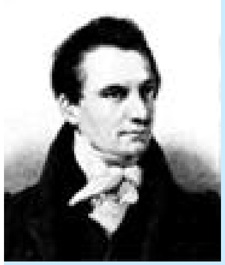


Combination of Hans Moravac + Larry Roberts + Gordon Bell
WordSize*ops/s/sysprice

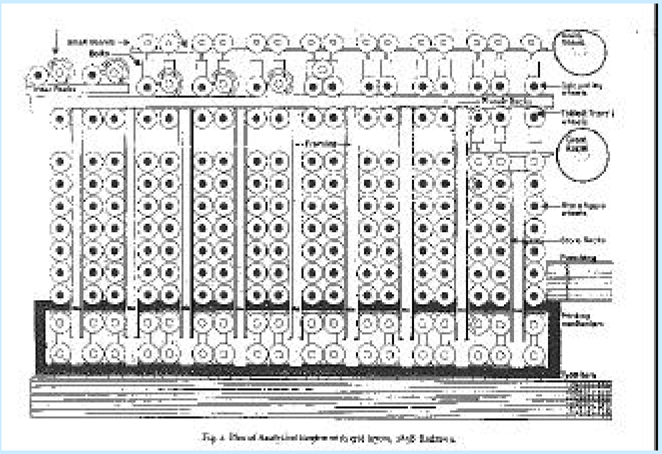


# Trouble-Free Appliances

- Appliance just works.     TV,  PDA, desktop, ...
- State replicated in safe place (somewhere else)
- If hardware fails, or is lost or stolen,
  replacement arrives next day (plug&play).
- If software faults,
  software and state refresh from server.
- If you buy a  new appliance, it plugs in and refreshes from the server (as though the old one failed)
- Most vendors are building towards this vision.
- Browsers come close to working this way.

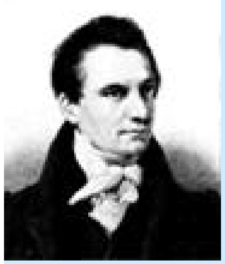


# Trouble-Free Systems



- Manager
  - Sets goals
  - Sets policy
  - Sets budget
  - System does the rest.
- Everyone is a CIO (Chief Information Officer)

9. Build a system
   - used by millions of people each day
   - Administered and managed by a ½ time person.
     - On hardware fault, order replacement part
     - On overload, order additional equipment
     - Upgrade hardware and software automatically.

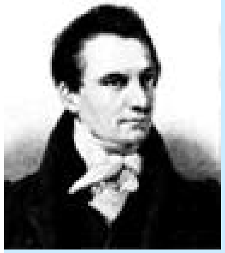


# Trustworthy Systems

- Build a system used by millions of people that

10. Only services authorized users
    - Service cannot be denied (can’t destroy data or power).
    - Information cannot be stolen.
11. Is always available: (out less than 1 second per 100 years = 8 9’s of availability)
    - 1950’s 90% availability,<br>Today 99% uptime for web sites,<br>99.99% for well managed sites (50 minutes/year)<br>3 extra 9s in 45 years.
    - Goal: 5 more 9s: 1 second per century.

– And prove it.



# 100 $ line of code?
# 1 bug per thousand lines?

- 20 $ to design and write it.
- 30 $ to test and document it.
- 50 $ to maintain it.

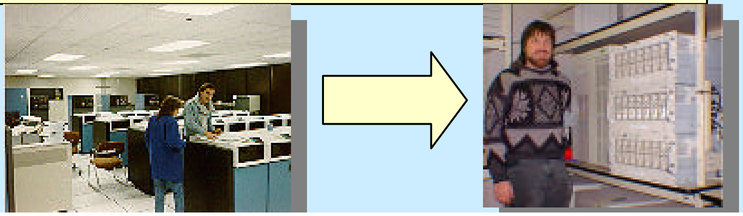
- The only thing in Cyber Space that is getting
  MORE expensive
  LESS reliable

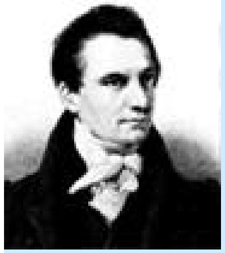
Solution so far:

- Write fewer lines
  High level languages
- Non Procedural

- Application generators:
  Web sites, Databases, ...
- Semi-custom apps:
  SAP, PeopleSoft,..
- Scripting & Objects
  JavaScript & DOM

•10x not 1,000x better
Very domain specific





# Automatic Programming

Do What I Mean (not 100$ Line of code!, no programming bugs)

The holy grail of programming languages & systems

12. Devise a specification language or UI
    1. That is easy for people to express designs (1,000x easier)
    2. That computers can compile.
    3. That can describe all applications (is complete).

- System should “reason” about application
  - Ask about exception cases.
  - Ask about incomplete specification.
  - But not be onerous.
- This already exists in domain-specific areas. (i.e. 2 out of 3 already exists)
- An imitation game for a programming staff.

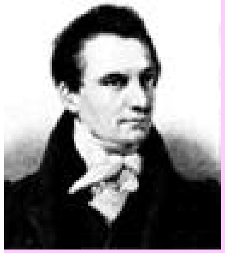
# Summary

- Invest in long-term research:
  - Creates new ideas
  - Teaches students
  - Governments should fund some of it.
- Long Term Research projects:
  - Human-computer interface:
    make computers easier to communicate with.
  - Organize, Summarize, and Analyze information
  - Auto-managing & auto-programming
- Paradoxically, many problems come back to machine intelligence (the Turing Test)

# The List (Red is Turing Complete)

1. Devise an architecture that scales up by 10^6.
2. The Turing test: win the impersonation game 30% of the time.
    a. 3.Read and understand as well as a human.
    b. 4.Think and write as well as a human.
3. Hear as well as a person (native speaker): speech to text.
4. Speak as well as a person (native speaker): text to speech.
5. See as well as a person (recognize).
6. Illustrate as well as a person (done!) but virtual reality is still a major challenge.
7. A copy-protection and payment scheme that protects IP owner and user.
8. Remember what is seen and heard and quickly return it on request.
9. Build a system that, given a text corpus, can answer questions about the text and summarize it as quickly and  precisely as a human expert.
10. Do 9 for Sounds: conversations, music.
11. Do 9 for Images: pictures, art, movies.
12. Simulate being some other place as an observer (Tele-Past) and a participant (Tele-Present).
13. Automatic Programming: Given a specification, build a  system that implements the spec. Prove that the implementation matches the spec. Do it better than a team of programmers.
14. Build a system  used by millions of people each day but administered  by a ½ time person.
15. Do 15 and prove it only services authorized users.
16. Do 15 and prove it is almost always available: (out less than 1 second per 100 years).

- **B**abbage's vision of computing has largely been realized. We are on the verge of realizing Bush's Memex. But, we are some distance from passing the Turing test. These three visions and their associated problems have provided long-range research goals for many of us. For example, the scalabilty problem has motivated me for several decades. This talk defines a set of fundamental research problems that broaden the Babbage, Bush, and Turing visions. They extend Babbage's computational goal to include highly-secure, highly-available, self-programming, self-managing, and self-replicating systems. They extend Bush's Memex vision to include a system that automatically organizes, indexes, digests, evaluates, and summarizes information (as well as a human might). Another group of problems extends Turing's vision to include prosthetic vision, speech, hearing, and other senses. Each problem is simply stated and each is orthogonal from the others, though they share some common core technologies.



# Automatic Programming

Do What I Mean (not 100$ Line of code!, no programming bugs)

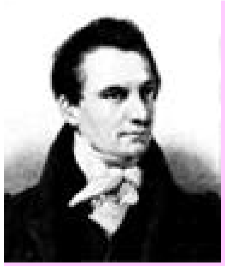
The holy grail of programming languages & systems

## 13. Automatic Programming:

- Given a specification, build a  system that implements the spec.
- Prove that the implementation matches the spec.
- Do it better than a team of programmers.

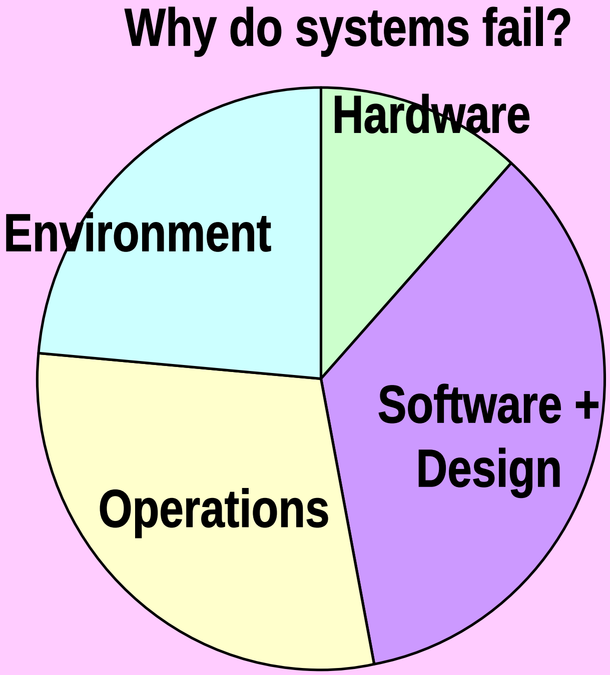
- Program Specifications are:
  - Difficult to write
  - Difficult to read
  - Often incomplete
- Humans translate Spec to Program
  - Fills in the gaps
  - Otherwise mechanical
  - Error prone and expensive
- Must start at a higher level

- Automatic programmer
  - System “discusses” problem with designer
  - System builds prototype
  - Customer evaluates prototype
- Its is a Turing test – imitate a programmer (staff).



# Why Do Systems Fail?

- Have to solve *all* these problems:
  - Hardware: redundancy and *replication software*
  - Operations: *self managing software*
  - Environment: redundancy and *fault-tolerance software*
  - Software: *and more software!* Software (& design) Faults (Heisenbugs)
- Mask all these faults



•Requires many new approaches

–Design diversity

–Geographic diversity

–Self-managing, self healing

# Extended Turing test (avatar)

- Judge, man, and  woman all in cyberspace
  - Audio/video/… chat room.
- Man has a life-like avatar (video projection).
- Computer has a life-like avatar
- Judge cannot distinguish man from computer.

# Software Laws

- Nathan's Law:
  - Software is a gas: it expands to fill the volume.
- Augustine's Law XVII:
  - Software is like entropy. It is difficult to grasp, weighs nothing, and obeys the Second Law of Thermodynamics; i.e. it always increases.

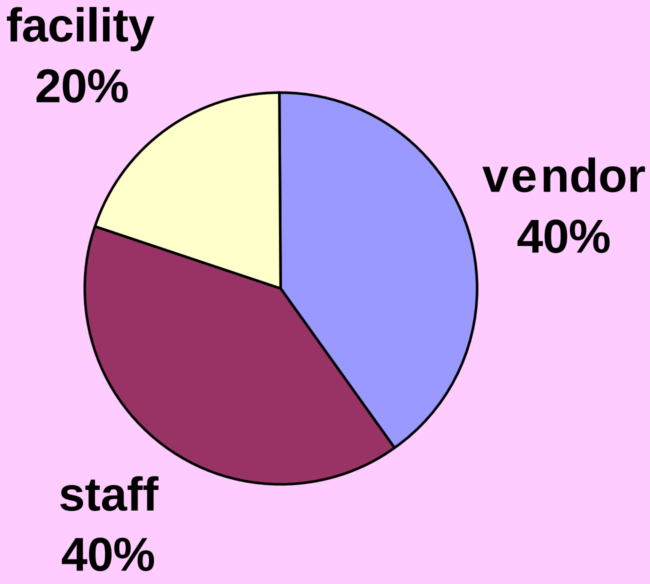
# So, what’s long-range research?

- Something that we do not know how to do.
- Something that venture capital will not fund.
  - i.e. More than 2 years to product.
- For systems work,
  something that will have clear benefit.
- In many cases,
  a problem that has defeated others:
  needs VERY DIFFERENT approach.

# Where does the money go?

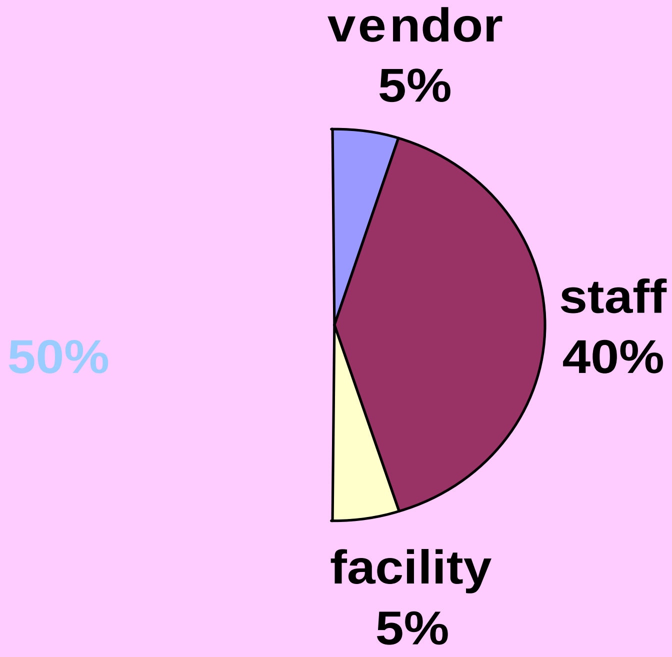
- **Old world**:

  Balanced system.



- **New World**

  Operations is 80% of cost



**In the new world, operations is dominant cost.**