Big Data
Yesterday, Today, and Tomorrow

John R. Mashey
Tuesday 08/18/15 for

IEEE Computer Society Santa Clara Valley Chapter
Speaker – John R. Mashey

Pennsylvania State University, 1964-1973, BS Math, MS/PhD Computer Science

Bell Labs 1973-1983, MTS → Supervisor, early UNIX
  – Programmer’s Workbench, shell programming, text processing, workload measurement/tuning in first UNIX computer center, UNIX+mainframe data mining apps, capacity planning/tuning

Convergent Technologies 1983-1984, MTS → Director Software
  – Compiler & OS tuning, uniprocessor/multiprocessor servers

  – System coprocessor, TLB, interrupt-handling; byte addressing(!), halfword instructions; ISA evolution, multiprocessor features, multi-page-size TLB, 64-bit
  – MIPS Performance Brief editor; a SPEC benchmarking group founder 1988
  – Hot Chips Conference committee … 1989-continuing

  – ccNUMA system architecture (NUMAf lex in Origin3000, Altix, still) … small to REALLY BIG
  – Performance issues in HPC, DBMS; technology forecasting
  – Evangelist, much work with sales and marketing, business development, strategy
  – By 1994, started using “Big Data” in modern sense, made it an SGI marketing theme

Typical Silicon Valley “Semi-retired” – occasional consulting for high-tech co’s, VCs
Technical advisory boards, lately for (stealth) parallel hardware or software co’s
Computer History Museum (www.computerhistory.org) Trustee
Ski; hike; bike; investigative reporting write articles & speak, blog on climate issues, tobacco cntrl
The Origins of ‘Big Data’: An Etymological Detective Story
By STEVE LOHR

Words and phrases are fundamental building blocks of language and culture, much as genes and cells are to the biology of life. And words are how we express ideas, so tracing their origin, development and spread is not merely an academic pursuit but a window into a society’s intellectual evolution.

Since I first looked at how he used the term, I liked Mr. Mashey as the originator of Big Data. In the 1990s, Silicon Graphics was the giant of computer graphics, used for special-effects in Hollywood and for video surveillance by spy agencies. It was a hot company in the Valley that dealt with new kinds of data, and lots of it. When I called Mr. Mashey recently, he said that Big Data is such a simple term, it’s not much a claim to fame. His role, if any, he said, was to popularize the term within a portion of the high-tech community in the 1990s. “I was using one label for a range of issues, and I wanted the simplest, shortest phrase to convey that the boundaries of computing keep advancing,” …

At the University of Pennsylvania, Mr. Diebold kept looking into the subject as well. … His most recent paper concludes: “The term Big Data, which spans computer science and statistics/econometrics, probably originated in the lunch-table conversations at Silicon Graphics in the mid-1990s, in which John Mashey figured prominently.”

http://en.wikipedia.org/wiki/Talk:Big_data
Overview

• Big Data – has my definition changed?

• Yesterdays – many, with some lessons
  – “Those who cannot remember the past are condemned to repeat it” George Santayana
  – “Big data” had many different labels, the sizes changed
  – “Stack wars” during each stage, both hardware and software

• Today – how did we get here
• Tomorrow – 30,000-foot view, issues

• Q&A (but ask clarification questions while slides up … may hold)
Big Data – “How has your definition changed from 1994?”

- Not at all!??
- My definition always had 2 ideas:
- Big Data was:
  - Beyond widely-available computers (Current)
    Memory capacity (Volume)
  - I/O Performance / Real-time issues (Velocity)
    Complex/difficult data – multimedia, 3D models, scientific data sets (Variety)
- Big Data was a moving target, problems evolved downward
  - Impossible within the state of the art, even with huge $$$
  - Big – needs large system or tight-coupled network of them
  - Nontrivial programming, sometimes much roll-your-own
  - Easy – plus a few local disks (or once upon a time, tapes)
  - Trivial – fits in memory of widely-available system
- Large 1970 mainframes could get 1MB memory (~1us), 8MB slow
  ~ <=Cache memory on current laptops
- Really, people have done Big Data for a long time, but buzzwords change
Big Data - 1890

- Early Big Data

Big Data – 1950

- IBM 026 Keypunch, 082 Sorter and 403 Accounting Machine (c.1950)*

Plugboard … “software” for the accounting machine


* But this sort of gear was still around in 1967. I did actually use a sorter at Penn State a few times when box of program cards got dropped.
Big Data ~1960s-

- IBM 360 Model 40 (c.1965)

24-bit address: 16MB memory maximum.

A “Big” mainframe was a 360/65, 67* or 75:  
Up to 1MB of main (magnetic) core storage.  
Optionally, up to 8MB of slower “Large Core Storage”  
360/40 -medium system with 4 tape drives in back,  
and 2 2311 disk drives, each max capacity = 7.2MB.  
By 1966, the first 2314 drives shipped, each 28MB.  
A system with 10 28MB drives was big.  

A rare IBM 2321 Data Cell ** nicknamed “noodle snatcher” or “washing machine” offered 400MB of random-access storage!!  
Ours worked OK, but many had mechanical trouble.  

1968: 360/67: first IBM 360 with virtual memory …  
CP-67/CMS → VM/CMS virtual machines … 40+ years  
SERVICE BUREAUS  
shared access to mainframes, otherwise unaffordable  
CENTRALIZED  
Stack wars from hardware up: big vendors

Image courtesy of Computer History Museum.  
www.computerhistory.org/collections/catalog/102618836

By 1960s, Big Data was 2400ft, 8-Track tape  
40MB max – 1600 BPI IBM 24xx  
In 1970s, IBM upped this:  
170MB max – 6250 BPI IBM 34xx, 1.25MB/sec
Big Data ~1970s

- IBM 370, virtual-memory upgrades from 360s
  - 1972- 370/168: typical high end: up to 8MB of fast memory

Not Big Data yet, but signs on the horizon…

- Minicomputers really got rolling - Digital Equipment Corporation
  - 1975 – PDP-11/70, up to 4MB – 48 users
  - 1975 – DEC RP04 disk drive ~92MB
  - 1977 - VAX 11/780, up to 8MB
  - 1977 – DEC RP06 disk drive ~178MB
  - Clustering to compete with real mainframes

- Microprocessors? Late in decade, “toys”
  - But personal computers, workstations got going

- Stack wars – algorithmic languages, OSs

Image courtesy of Computer History Museum.
www.computerhistory.org/collections/catalog/102685442
Big Data ~1970s – 1980s

• Bell Labs (1973-1983), 1M+ person Bell System, 25,000 people in R&D
  – First half: Programmer’s Workbench/UNIX
    » “Small is Beautiful and Other Thoughts on Programming Strategies” 😊
    » Use tools, existing software, scripting languages
    » Small teams, fast iterations / prototyping, move quickly, get user feedback (sound familiar?)
  – Big Data
    » Track every telephone pole, cable, junction box, geography, trouble reports, squirrels, guns
    » ~400 operations support systems, not all compatible
  – Charge people for $0.10 phone calls: Murray Hill, NJ Building 5 analyzed call records
  – Loop Maintenance Operations System (LMOS)
    » IBM Mainframe with larger database, with triplexed minicomputers to support calls to 411
    » CRAS – backend data mining system, ACE expert system offshoot (BTL 1st)
    » LMOS transaction database work ➔ Tuxedo ➔ Novell ➔ BEA ➔ Oracle

– The “cloud” found in many internal talks = Bell System
  » AGS (BDN, Net/4, Net/1000) ➔ processing in the network, dumb terminals at edge, $1B
  » Attempt to build a packet-switched “Internet Cloud” in 1970s … too early
Big Data ~1980s

- Teradata – parallel data warehouses, special hardware (started in 1970s)
  - 1983 – first Beta system, 1991 NCR ➔ AT&T, Teradata ➔ NCR, later spun out
  - Finally started using “Big Data” term in 2010 … but classic Big Data company from start

- 1987 RAID – UC Berkeley, Patterson, Gibson, Katz

Not “Big Data” yet, but more signs on the horizon…
- Network stack wars: Networks, DARPA funded UCB
  - 1983 – 4.2BSD, with TCP/IP
- RDBMS wars – Oracle, Informix, Sybase, etc, etc
- UNIX stack version wars, by vendor and camp
- Distribution wars: workstations, client-server, PCs, networks, thin-clients
- Microprocessor wars of late 1980s, early 1990s
  - Multiprocessors for bigger systems, rapid end for most minicomputer companies
  - 1988 First 64-bit microprocessor, MIPS R4000 ➔ 1991
- More stack wars – window systems
Big Data ~1990s (SGI)

- Internet growing, WWW, Multimedia
- Browser wars
- Linux on the horizon

- 1992 – 64-bit micro-based systems
  - SGI Crimson, early 1992, still running 32-bit OS, no new 32-bit-only designs
  - DEC Alpha systems, late in year, 64-bit-only (plausible)

- 1994 - SGI
  - Large systems running 64-bit IRIX, with 64/32-bit user programs
  - 1993’s Challenge XL got OS upgrade, up to 36p or 16GB memory
  - XFS – Full 64-bit UNIX file systems, journaled, for serious Big Data
    » Later contributed to Linux, along with scalability improvements
  - Customers could finally just recompile and use >4GB of memory in one program
- “The Long Road to 64 Bits – Double, Double, Toil and Trouble”
  http://queue.acm.org/detail.cfm?id=1165766
“HW, WW, SS” Traditional SGI

SGI – SC’96
Supercomputing show
November 1996
“HW, WW, SS” – New Markets

- User Interface
- Digital media
- "Publishing"
- Computer Aided Software Engineering
- Database Management, Data Mining
- New visualization ("data", not scientific)
“Hardware, Wetware, Software”

“UNIX System – I know this.”
-Jurassic Park

Video1
Video2
Lower Response Times ➔ Changes Applications
Big Data @ SGI ~1995

**CAPITALIZE On Database Success in '94 and '95**

<table>
<thead>
<tr>
<th>Corporation</th>
<th>Industry</th>
<th>Database System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corpovon Oil</td>
<td>Oil and Gas</td>
<td>Oracle</td>
</tr>
<tr>
<td>Temda</td>
<td>Pharmaceuticals</td>
<td>Oracle</td>
</tr>
<tr>
<td>Moen</td>
<td>Manufacturing</td>
<td>Oracle</td>
</tr>
<tr>
<td>Dacom</td>
<td>Telecom</td>
<td>Oracle</td>
</tr>
<tr>
<td>Chosun Daily News</td>
<td>Media</td>
<td>Sybase, Basis+</td>
</tr>
<tr>
<td>Korean Defense Agency</td>
<td>Government</td>
<td>Basis+</td>
</tr>
<tr>
<td>NOWCOM</td>
<td>Telecom</td>
<td>Informix</td>
</tr>
<tr>
<td>Korea Electronic</td>
<td>Manufacturing</td>
<td>Oracle</td>
</tr>
<tr>
<td>KSFC</td>
<td>Financial</td>
<td>Oracle</td>
</tr>
<tr>
<td>Norwegian Stock Exchange</td>
<td>Financial</td>
<td>Oracle</td>
</tr>
<tr>
<td>Narrae Mobile Telecom</td>
<td>Telecom</td>
<td>Informix</td>
</tr>
<tr>
<td>IsComp / Fry Steel</td>
<td>Manufacturing</td>
<td>Oracle</td>
</tr>
<tr>
<td>Dept of Commerce</td>
<td>Government</td>
<td>Informix</td>
</tr>
<tr>
<td>Eastman Kodak</td>
<td>Manufacturing</td>
<td>Sybase</td>
</tr>
<tr>
<td>GDE</td>
<td>Government</td>
<td>Sybase</td>
</tr>
<tr>
<td>NASA Goddard</td>
<td>Government</td>
<td>Sybase</td>
</tr>
<tr>
<td>GFDL</td>
<td>Transportation</td>
<td>Sybase / Homegrown</td>
</tr>
<tr>
<td>SMC</td>
<td>Manufacturing</td>
<td>Informix</td>
</tr>
<tr>
<td>Rubbermaid</td>
<td>PTC proprietary</td>
<td>Sybase</td>
</tr>
<tr>
<td>Automatic</td>
<td>Government</td>
<td>Oracle</td>
</tr>
<tr>
<td>Sojo Mutual Credit Bank</td>
<td>Financial</td>
<td>Oracle</td>
</tr>
<tr>
<td>Northern Petro Search</td>
<td>Oil and Gas</td>
<td>Oracle</td>
</tr>
<tr>
<td>AT&amp;T Bell Labs</td>
<td>Telecom</td>
<td>Oracle</td>
</tr>
<tr>
<td>SISE</td>
<td>Government</td>
<td>Oracle</td>
</tr>
<tr>
<td>PEMEX</td>
<td>Oil and Gas</td>
<td>Oracle</td>
</tr>
<tr>
<td>Banco Banobras</td>
<td>Financial</td>
<td>Oracle</td>
</tr>
<tr>
<td>Alcaldia de Cali</td>
<td>Government</td>
<td>Oracle</td>
</tr>
</tbody>
</table>

2 Chall L, 1 S
Chal L
Chal L, 2 DM
Chal XL
2 Chall XL
Chal L
Chal L
2 Chall L
Chal L
Chal L
2 Chall L
Chal L
Chal L
2 Chall L
Chal XL
Chal L
Chal L
2 Chall L
Chal XL
Chal L
Chal L
5 Chall DM
Chal L, 2 DM
Chal XL
Chal XL
2 Chall DM
Chal L
Chal L
2 Chall DM
Chal DM
3 Chall L, 2 M
Chal L
Chal L
Chal L
Big Data @ SGI ~1995

<table>
<thead>
<tr>
<th>Boeing</th>
<th>Manufacturing</th>
<th>Oracle</th>
<th>2</th>
<th>Chal Ls</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTE-TSI</td>
<td>Telecommunications</td>
<td>Informix</td>
<td>2</td>
<td>Chal XL, 4L</td>
</tr>
<tr>
<td>Salomon Bros.</td>
<td>Brokerage and Analytics</td>
<td>Sybase</td>
<td>20</td>
<td>Chal Ls</td>
</tr>
<tr>
<td>Wang–New Zealand Telecom</td>
<td>Telecommunications</td>
<td>Oracle/Ingres</td>
<td>Chal XL, 2 Ls</td>
<td></td>
</tr>
<tr>
<td>Wang – Tasman Forestry</td>
<td>Government</td>
<td>Sybase</td>
<td>Chal DM, 5 Ss</td>
<td></td>
</tr>
<tr>
<td>Northwest Natural Gas</td>
<td>Utility</td>
<td>Sybase</td>
<td>3</td>
<td>Chal Ls, 34 Ms</td>
</tr>
<tr>
<td>Time Warner</td>
<td>Entertainment</td>
<td>Oracle</td>
<td>5</td>
<td>Chal Ls</td>
</tr>
<tr>
<td>Norcon Energy Resources</td>
<td>Oil and Gas</td>
<td>Oracle</td>
<td>Chal L</td>
<td></td>
</tr>
<tr>
<td>International Billing Services</td>
<td>Billing and Mailing</td>
<td>Home Grown</td>
<td>5</td>
<td>Chal L</td>
</tr>
<tr>
<td>Goldman–Sachs</td>
<td>Brokerage and Analytics</td>
<td>Oracle</td>
<td>3</td>
<td>Chal Ls</td>
</tr>
<tr>
<td>Eagle–New Zealand IRO</td>
<td>Government</td>
<td>Progress</td>
<td>2</td>
<td>Chal Ls</td>
</tr>
<tr>
<td>Stats Canada</td>
<td>Gov’t</td>
<td>Sybase/Oracle</td>
<td>Chal XL, M, S</td>
<td></td>
</tr>
<tr>
<td>Wang–Energy Direct</td>
<td>Utility</td>
<td>Informix</td>
<td>Chal L</td>
<td></td>
</tr>
<tr>
<td>PRC</td>
<td>Telecom and Financial</td>
<td>Oracle</td>
<td>8</td>
<td>Chal L</td>
</tr>
<tr>
<td>Tandem</td>
<td>Automotive</td>
<td>Oracle/Sybase</td>
<td>Chal M/L/XL</td>
<td></td>
</tr>
<tr>
<td>Reynolds &amp; Reynolds</td>
<td>Telecom</td>
<td>Vmware</td>
<td>Chal S/DM</td>
<td></td>
</tr>
<tr>
<td>Ameritech Services</td>
<td>Aerospace</td>
<td>Progress</td>
<td>Chal XL/L</td>
<td></td>
</tr>
<tr>
<td>Grumman Tech Services</td>
<td>Gov’t Aerospace</td>
<td>Oracle</td>
<td>Chal L</td>
<td></td>
</tr>
<tr>
<td>NASA Ames</td>
<td>Government</td>
<td>Sybase</td>
<td>4</td>
<td>Chal L</td>
</tr>
<tr>
<td>E-Systems</td>
<td>Telecommunications</td>
<td>Sybase</td>
<td>9</td>
<td>Chal L</td>
</tr>
<tr>
<td>AT&amp;T Bell Labs</td>
<td>Gov’t Aerospace</td>
<td>Ingres</td>
<td>Chal XL</td>
<td></td>
</tr>
<tr>
<td>NASA Langley</td>
<td>Semiconductor Mfg</td>
<td>Home Grown</td>
<td>10</td>
<td>Chal L</td>
</tr>
<tr>
<td>Lam Research Center</td>
<td>Transportation</td>
<td>Oracle</td>
<td>4</td>
<td>Chal L</td>
</tr>
<tr>
<td>Condor</td>
<td>Gov’t Aerospace</td>
<td>Home Grown</td>
<td>9</td>
<td>Chal L</td>
</tr>
<tr>
<td>NASA Marshall</td>
<td>Defense</td>
<td>Oracle</td>
<td>Chal XL</td>
<td></td>
</tr>
<tr>
<td>US Army – Aberdeen</td>
<td>Financial</td>
<td>Oracle/Informix/Ingres</td>
<td>2</td>
<td>Challenge L</td>
</tr>
<tr>
<td>GFTA</td>
<td></td>
<td>Home Grown</td>
<td>2</td>
<td>Chal XL 3 L</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>Chal L, XL M</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>Challenge XL</td>
</tr>
</tbody>
</table>

Techviser

Big Data – Yesterday, Today and Tomorrow
Big Advantages from Big Tools for Big Data

America On-Line
- OLTP systems running email, news
  - 100 mid-range servers
  - 70 large servers
  - several hundred Sybase databases
- Web hosting
  - 200 Challenge-S servers
- Large Redbrick datawarehouses
  - ~1TB each

AT&T
- AT&T Business marketing datawarehouses/marts
  - 1–800–COMPARE etc.
  - 30+ Oracle databases
- Large data analysis systems in AT&T Labs
  - 1TB Server, XFS and Oracle
  - Graphics Supercomputers
- Billing fulfilment systems
  - 3 servers with IBS software and Oracle

Big Data @ SGI ~1997
Big Data @ SGI ~1997

Big Advantages from Big Tools for Big Data

Alliance Data Systems
- The Limited department stores
- 2TB Informix marketing data warehouse

People’s Bank
- Solution by Experian (TRW Credit)
- Credit card marketing data warehouse
  - Phase 1 30MM name 1TB Oracle database
  - Phase 2 60MM name 2 TB Oracle database

Sygnent Bank
- ~200GB Oracle marketing data warehouse

Salomon Brothers
- “Yield Book” portfolio management system
  - ~30+ Sybase servers, HPC servers

Morgan Stanley
- Global risk management system
Big Advantages from Big Tools for Big Data

Telecom New Zealand
- OLTP National Faults System
  - 2 SGI servers replaced 28 IBM RS/6000’s!
  - 2,000+ users
- HR and payroll systems
- Next generation sales and customer support systems
  - Oracle consulting project
  - Tuxedo based
  - AS/400 Integration
- Oracle marketing datawarehouse
  - Origin 2000, >0.5 TB data, 1.6 TB disk
  - Teradata replacement

National Security Applications
- Hundreds of large servers running data storage and HSM
- Largest Oracle DBMS demonstrated
Big Data @ SGI ~1998

Big Data ...
and the Next Wave of **InfraStress**

John R. Mashey
Chief Scientist, SGI

Technology Waves:
NOT technology for technology’s sake
IT’S WHAT YOU DO WITH IT
But if you don’t understand the trends
IT’S WHAT IT WILL DO TO YOU

What is Strategic Business Analysis

Solutions that allow customers to gain extreme competitive advantage

Better understand and manage:
- Internal operations and resources
- Customer Behavior
- External environments

- Database marketing
- Financial analytics
- Fraud management
- Bioinformatics
- Supply chain optimization
- Law enforcement
- Media data management
- Schedule optimization
### SBA Customers

**Communications**
- Ameritech
- AOL
- BellSouth
- Cable & Wireless
- GTE-TSI
- Pacific Bell

**Government**
- Army & Air Force Exch. Svc
- NASA
- NCSA
- State of Texas
- US Census Bureau

**Manufacturing**
- Chevron (Corp. Fin.)
- Ericsson
- Ford Motor
- GM - Buick
- GM - Cadillac
- LAM Research

**Other**
- Beneficial
- Columbia Healthcare
- Condor Air
- Prime Option
- WFN

**Financial Services**
- Allstate Insurance
- AT&T Universal Card
- Banamex (Mexico)
- Bank of Canada
- Bear Steam
- Capital One
- Chase Bank
- Financial Net
- First Union
- Goldman Sachs
- John Hancock
- LCM
- Member Fin Svc
- Morgan Stanley
- Natwest Markets
- Novus
- People’s Bank
- Salomon Brothers
- Sumito visa

**Retail/CPG**
- Charles Vogele
- E. LeClerc
- Hannaford Bros
- Nordstrom
- Procter and Gamble
- QVC
- Rimi Stores
- Unilever

**Chemical/Sciences**
- Bristol Myers Squibb
- Glaxco
- Merck
- Pfizer
- Smithkline Beecham
- Zeneca

**Service Bureau**
- Alliance Data Systems
- EDS
- Fannie Mae
- Epsilon
- Experian
- International Billing Svcs
- May and Speh
- MetroMail
- Neo Data

---

*Page 12*
Big Data @ SGI ~1998-

SGI in SBA Today

✓ Over 40% of cellular fraud detection is handled on Silicon Graphics Servers
✓ Seven out of top ten investment banking firms run risk analysis on Silicon Graphics servers
✓ Over 6500 MineSet visual and analytical data mining licenses sold in 8 quarters
✓ MineSet is installed in 13% of data mining user base
✓ Over 25,000 CC-NUMA servers installed
✓ Ranked #4 in mid-range UNIX Servers (IDC)

October 16, 1998
Big Data Present – How Did We Get Here?

- 1994 Beowulf technical PC clusters (not Big Data, but sign on horizon)
- Virtual machines … ~ IBM VM from 1970s, VMware founded 1998
- Web companies like Google, etc
- 2003 AMD 64-bit X86, 2004 Intel … finally
- Map-Reduce, Hadoop – software infrastructure did for relevant apps
  - What parallel programming tools did in 1990s for technical codes
  - What WWW did for making Internet more accessible
- Amazon Web Services, etc … echo of 1970s service bureaus
- Open-source sharing, GitHub, etc.
Retrospective – “Open Source”

• “Open Source” is most recent term for “ancient” practice
  1948 – David Wheeler invents subroutines for EDSAC@ Cambridge
  1952 – John von Neumann gives designs for Princeton IAS to anyone who asked
  1955 – IBM SHARE User’s Group founded; user groups trade code
  1961 – DECUS (Digital Equipment Corporation) user group founded
  1960s – IBM HASP (mainframe OS code, user-modified)
    “Should old Chuck Forney be forgot, and HASP songs sung no more.” *
  1960s – IBM S/360 vastly increases set of compatible systems, code-trading
    Penn State ASSIST (Mashey & others), 1970- … still running 30+ years later!
  1970s – UNIX “open source” within Bell Labs
  1970s – UNIX licensed to universities, government, “as is, don’t call us”
  1970s – John Lions “Commentary on UNIX, with Source Code”
  1970s – Berkeley UNIX, Ken Thompson, DARPA $, Internet
    ➔ Software Tools User’s Group (STUG) to get UNIXy code on other systems
  1979 – UNIX V7 released – (reasonably) portable OS
  1985 – Free Software Foundation (UNIX commands, especially GNU C)
  1991 – Linux (kernel); later Apache, etc, etc.

• Local libraries ➔ magnetic tapes ➔ UUCP ➔ Internet ➔ Web, GitHub, etc
• Local groups ➔ vendor-based groups ➔ large expansion for public

* Sung to “Auld Ang Sang”; Forney was Asst Director of PSU Computer Center
Big Data Future – 30,000-foot view

- Hardware discontinuities always cause turmoil, then industry settles
- Fairly predictable to happen, straightforward
  - Moore’s Law for CMOS still gets a few more turns, getting very hard
  - Clock-rate stalled → multi-core, parallel programming more important
  - Power/heat increasingly important
  - Disks keep getting denser, but bandwidth less so, seek/rotation times: no
  - Network bandwidths keep improving*
- Fairly predictable to happen, effects may be surprising
  - Internet-of-things, wireless sensor networks
  - Personal graphics, see Vernor Vinge, “Rainbow’s End”
  - Flash as more than SSD?
    - Big changes whenever change to memory hierarchy
    - Many software issues … 4KB pages (as per S/360) are bad news
Big Data Future – Stack wars

- Another period of turmoil in software in this area
- Tall, wide software stacks via multiple vendors, open source
  - Long ago, the whole stack was little more than Simple OS, compiler/assembler/linker, a few libraries
  - Now, big stacks … and do you trust everything below you?
- Creating software on top of stack
  - Assess choices carefully
  - As in 1980s, Russian Roulette with a few years’ delay
- Software inside a stack
  - Best technology does not always win
  - Alliances, partners really matter
  - Customers help!
- Anything that gets enough of market lasts a long time … last slide
Retrospective … Future

• John R. Mashey, “Languages, Levels, Libraries, and Longevity”

‘In 50 years, we’ve already seen numerous programming systems come and (mostly) go, although some have remained a long time and will probably do so for: decades? centuries? millennia? …

For the far future, Vernor Vinge’s fine science-fiction novel, A Deepness in the Sky, rings all too true. The young protagonist, Pham, has joined a starship crew and is learning the high-value vocation of “programmer archaeologist,” as the crew’s safety depends on the ability to find needed code, use it, and modify it without breaking something. He is initially appalled at the code he finds:

“The programs were crap…Programming went back to the beginning of time…There were programs here that had been written five thousand years ago, before Humankind ever left Earth. The wonder of it—the horror of it…these programs still worked…down at the very bottom of it was a little program that ran a counter. Second by second, the Qeng Ho counted from the instant that a human had first set foot on Old Earth’s moon. But if you looked at it still more closely… the starting instant was actually about fifteen million seconds later, the 0-second of one of Humankind’s first computer operating systems…”

“We should rewrite it all,” said Pham.

“It’s been done,” said Sura.

“It’s been tried,” corrected Bret…“You and a thousand friends would have to work for a century or so to reproduce it… And guess what—even if you did, by the time you finished, you’d have your own set of inconsistencies. And you still wouldn’t be consistent with all the applications that might be needed now and then…”

“The word for all this is ‘mature programming environment.’”
Extra
Money Can Buy Bandwidth, but Latency is Forever
Big Data @ SGI ~1998

SBA Application Framework

- Data Sources:
  - OLTP
  - Purchased Data

- Data Warehouse

- Analysis:
  - MQE/Query
  - Statistical Analysis
  - OLAP
  - Data Mining

- Action Plan Definition:
  - Mathematical Models
  - Data Based Models
  - Optimization

- Action Execution:
  - Marketing Plans
  - Compliance Programs

- Monitoring:
  - Campaign Manager
  - Monitor

Timeliness of solution is essential

October 16, 1998
### SGI Value Add: Mapping Technology to DW Needs

<table>
<thead>
<tr>
<th>Business Problem</th>
<th>Data Warehouse</th>
<th>SGI Origin 2000 Advantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple Data Sources</td>
<td>Large amounts of historical data</td>
<td>Up to 712 TB of Disk Storage Fiber Channel</td>
</tr>
<tr>
<td>- Marketing</td>
<td>Frequent Batch Updates</td>
<td></td>
</tr>
<tr>
<td>- Finance</td>
<td>Network Intensive</td>
<td></td>
</tr>
<tr>
<td>- Sales</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Legacy Loading</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manipulating large amounts of data</td>
<td>Separate from OLTP</td>
<td>FastEthernet, HIPPI, ATM, FDDI</td>
</tr>
<tr>
<td>Complex data joins</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fast access to data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Getting useful insight faster</td>
<td>Heavy Reads</td>
<td>Incredible I/O capacity &amp; bandwidth</td>
</tr>
<tr>
<td>Information availability</td>
<td>RDBMS table scans</td>
<td>Origin 8P 6.4 GB ⇒</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Origin 128P 102.4 GB ⇒</td>
</tr>
<tr>
<td></td>
<td>Analyzing the data (OLAP/Data Mining)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPU Intensive</td>
<td>Memory capacity &amp; bandwidth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Origin 8P 16 GB @ 3.2 GB/s ⇒</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Origin 128P 256 GB @ 51.2 GB/s ⇒</td>
</tr>
<tr>
<td></td>
<td>Data Availability</td>
<td>Scales from 1 to 128 CPUs</td>
</tr>
<tr>
<td></td>
<td>Reliability, backup, restore</td>
<td>1 TB backup in less than 1 hour</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 TB restore in less than 90 min.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>99% uptime</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HANS</td>
</tr>
</tbody>
</table>

October 16, 1998
For the 2002 BSDcon, I grabbed talks from 30 years ago, and used (images of) the original foils for authenticity, to help see what's changed and what's the same.*

The first part, "Small is Beautiful and Other Thoughts on Programming Strategies," was first used in 1977, and was later given many times as Association for Computing Machinery (ACM) National Lectures.

I was working on the Programmer's Workbench flavor of UNIX, and we'd had great success in making UNIX available to much wider audiences of software engineers targeting both UNIX-related and non-UNIX environments. We were strong believers in UNIX philosophies of tool-building and -using, and keeping software teams small during an era when there was strong emphasis on methodologies and large teams that were anything but lightweight. This talk was the result, and was considered somewhat radical at the time.

Scripting languages, development environments, “agile development”

* I still have the original foils, but they’re starting to wear out, and actually, old overhead projectors have started to disappear in favor of computers…. Originals were UNIX troff + hand-drawn graphics … not PowerPoint!

From: http://www.usenix.org/events/bsdcon/mashey_small, Thanks USENIX!
Evolution and Entropy

"In all cases observed in nature there is a tendency for processes to proceed toward a state of greater disorder."

PROGRAM EVOLUTION — Belady and Lehman

- Law of continuing change — a system that is used undergoes continuing change until it is judged more cost effective to freeze and recreate it.
  - Variety generated by the desire to perfect — continuing enhancement
  - Variety generated by imperfection — continuing maintenance
  - The result of continued evolution — structural complexity

- Law of increasing unstructuredness (entropy) — the entropy of a system increases with time, unless specific work is executed to maintain or reduce it.
How Things Get Complex

"Assigning repair responsibility to the "greenhorns" is of course the greatest fallacy of all."
— Belady and Lehman, in [BEL77A]

"Design maintenance programmers require a higher level of experience than that required for original design. Historically, management tended to place less experienced programmers in design maintenance. This is a costly and dangerous mistake."
— E. B. Daly (GTE/Automatic Electric), in [DAL77A]

"The simplest things, which only fifty years ago one could do without difficulty, cannot get done any more."
— E. F. Schumacher, in [SCH73B]

PARKINSON'S LAW "Things expand to limits"

GAMMON'S "BLACK HOLES"

PROBLEM IS: OK
SIMPLE COMPLEX

SOLUTION IS: OK
SIMPLE COMPLEX

Breakthrough
EVOLUTION AND ENTROPY

"In all cases observed in nature there is a tendency for processes to proceed toward a state of greater disorder."

PROGRAM EVOLUTION -- Belady and Lehman

- Law of continuing change -- a system that is used undergoes continuing change until it is judged more cost effective to freeze and recreate it.

  - Variety generated by the desire to perfect -- continuing enhancement
  - Variety generated by imperfection -- continuing maintenance
  - The result of continued evolution -- structural complexity

- Law of increasing unstructuredness (entropy) -- the entropy of a system increases with time, unless specific work is executed to maintain or reduce it.
How Things Get Complex

“Assigning repair responsibility to the "greenhorn" is of course the greatest fallacy of all.”
— Belady and Lehman, in [BEL77A]

“Design maintenance programmers require a higher level of experience than that required for original design. Historically, management tended to place less experienced programmers in design maintenance. This is a costly and dangerous mistake.”
— E. B. Daly (GTE/Automatic Electric), in [DAL77A]

“The simplest things, which only fifty years ago one could do without difficulty, cannot get done any more.”
— E. F. Schumacher, in [SCH73B]

PARKINSON’S LAW (“Things expand to limits”)

GAMMON’S “BLACK HOLES”

PROBLEM IS:

SOLUTION IS:

SIMPLE

Breakthrough

COMPLEX

OK

OK

VG. NO. 31
Featuris Creepis (Baby) later addition
Featuris Creepis (Adult)

later addition
Miscellaneous

• Code is malleable, invisible, more art than science
  – Yes, but much more visible than it used to be (open source + Web)
  – There has long been some (but not much) science in software engineering,
    there has always been art, [or good taste]
    there is more good engineering than there used to be

• Code is political, often instantiates rules invisibly
  – Yes, see Ravenflow, we still really need requirements ➔ code (automagically)
  – Business-English use cases converted to Visio charts by hand, ugh.
  – But, much code is now at high-level and visible
  – What’s an Excel spreadsheet? In the “old days”, it would have been FORTRAN

• Access privileges, work-arounds, trapdoors, audits
  – Humans have always been the weakest links
  – Overly-simple system is not sufficiently flexible
  – Overly-complex system is too much trouble, generates loopholes and bad behavior

• Code may be ephemeral … but actually has amazing longevity
  – Lasts far longer than hardware!
Use Existing Tools

USE EXISTING TOOLS ["Stop rebuilding oscilloscopes"]

- Earlier start, chance to avoid leapfrog
- Help keep product source small
- Work at highest level possible
- Tools available
  - Parser generators, compiler-compilers
  - Utility programs
  - Command languages used as programming languages
  - Existing code
- Uniform support facilities exist already, support and target can be separated
Silicon Valley Waves

WAVES OF INNOVATION


Techviser

Big Data – Yesterday, Today and Tomorrow