Why Free-Libre / Open Source Software (FLOSS)? Look at the Numbers!

David A. Wheeler

November 15, 2005

http://www.dwheeler.com/numbers
http://www.dwheeler.com/oss_fs_why.html

This presentation contains the views of the author and does not necessarily indicate endorsement by IDA, the U.S. government, or the U.S. DoD.

Outline of Quantitative Information on FLOSS*

Quantitive measures justify considering FLOSS

- **Background**
- **Quantitative measures**
  - **Market Share**
  - **Reliability**
  - **Performance**
  - **Scalability**
  - **Security**
  - **Total cost of ownership**
- **Non-quantitative**
- **Conclusions**

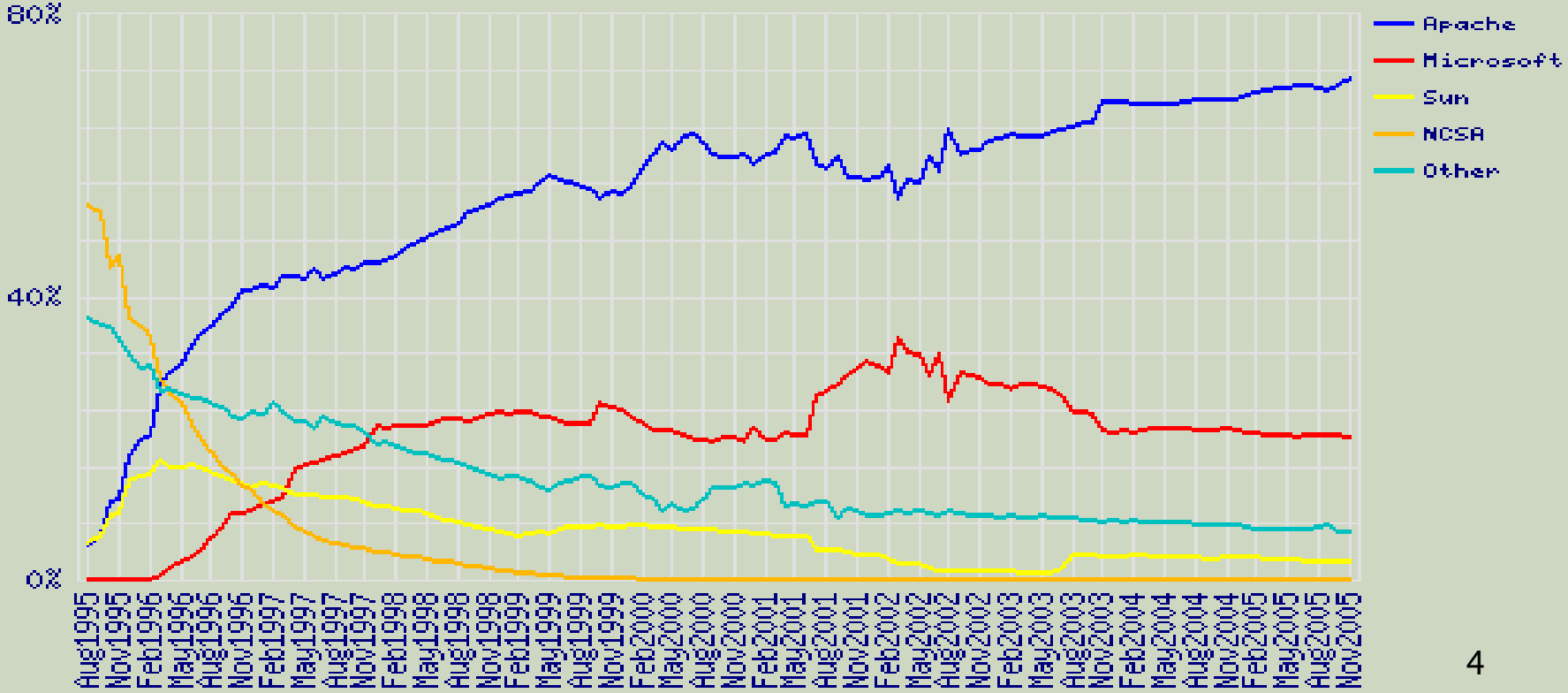
*FLOSS: Free-Libre / Open Source Software / Free Software; aka Open Source Software (OSS), Free Software (FS), OSS/FS, Libre or Livre Software, FOSS

Background

- In 2000, many claims about FLOSS, yet their advocates gave little evidence
- Investigated & found there *was* evidence
- Collection now widely-referenced
  - California Performance Review, 2004
  - Google “open source software” #5
- Challenges:
  - Vendor-funded studies (conflict of interest)
  - Some proprietary licenses forbid speech
- Numbers can't prove “always better”

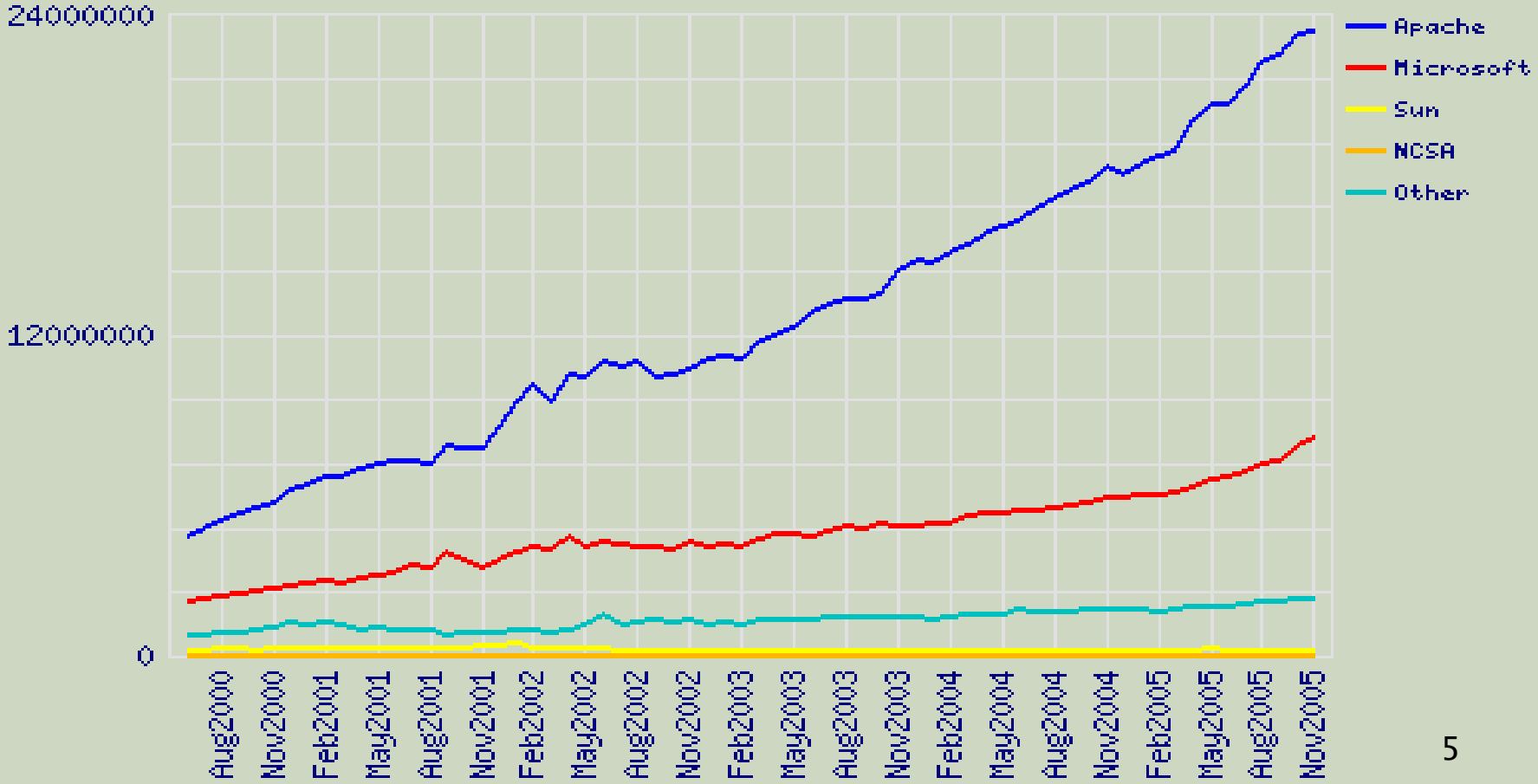
Market Share: Web Servers

- **FLOSS dominates web serving.**
  November 2005: Apache 70.98%, IIS 20.24%
  [Netcraft survey of 74,572,794 sites]



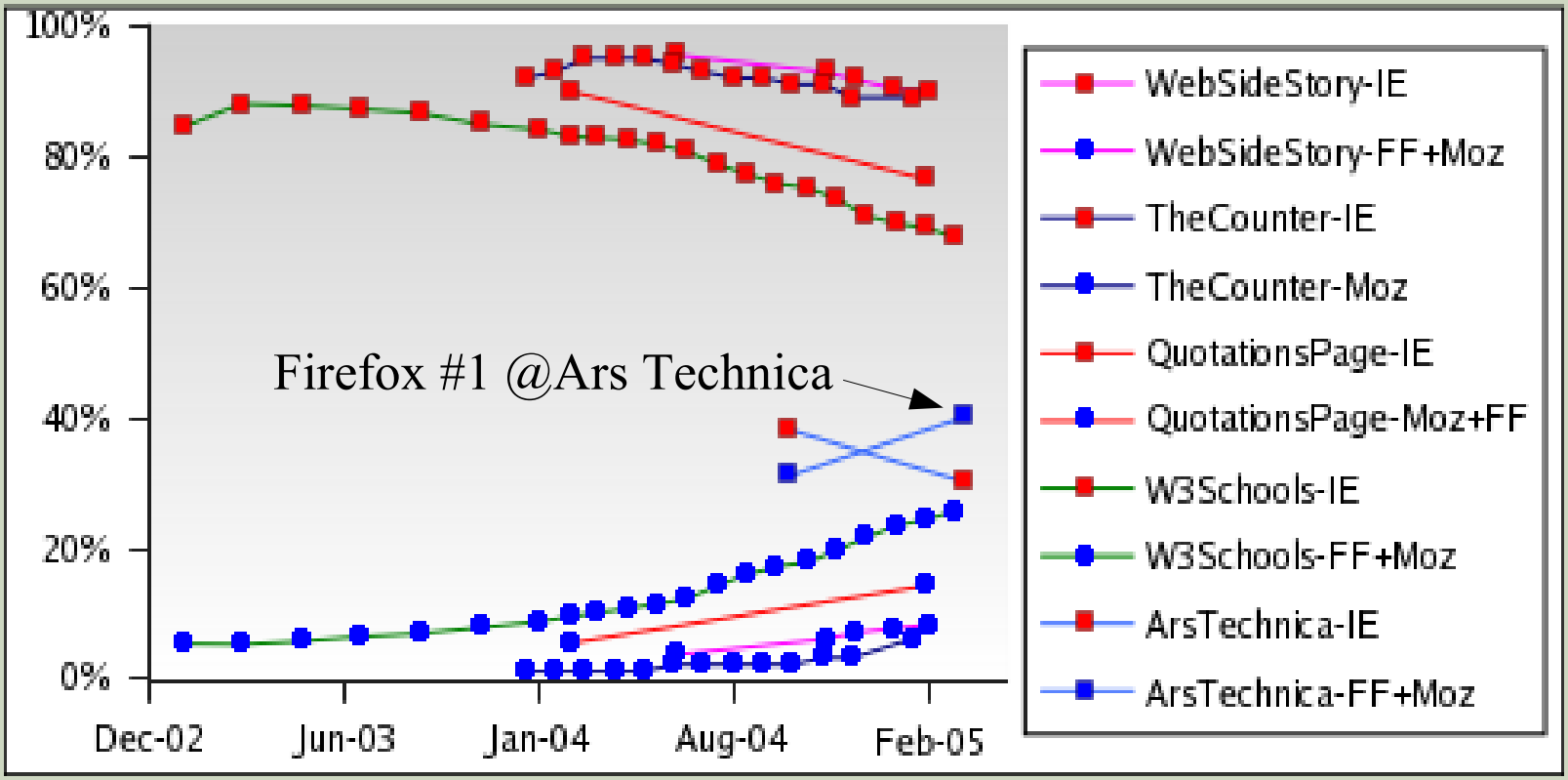
Market Share: Web Servers

- **Active Sites: Apache 69.36%, IIS 24.31% [Netcraft]**



Web Browsers: Growing Fast

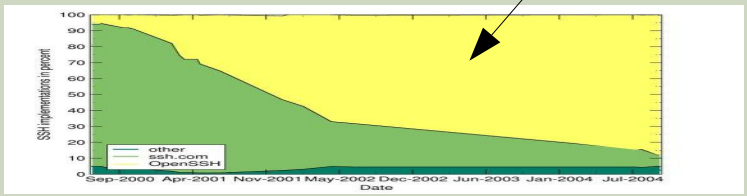
- **Mozilla/Firefox use growing, esp. among web/ technical community (who make web content!)**



Other Market Share Examples

- GNU/Linux #2 server OS sold 99, 00, 01 (24%, 27%, 25%)
- DNS: bind supports 95% of reverse-lookups [Manning]
- PHP #1 server-side scripting language [Netcraft]
- Sendmail #1 Email server [Bernstein]
  - Sendmail 42%, Microsoft Exchange 18%
- OpenSSH #1 SSH (87.9% Sep04)
  ~5% Summer 2000,
  50% November 2001 [scanssh]

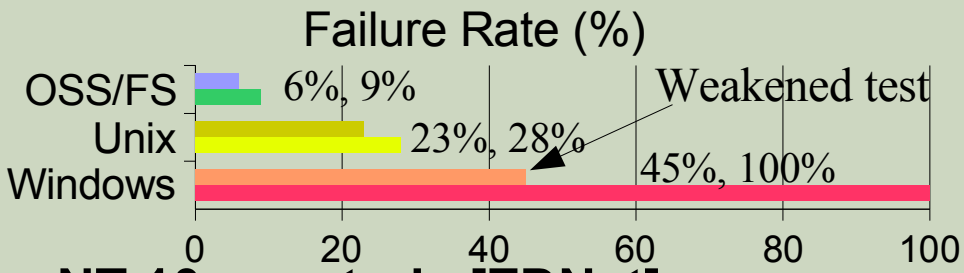
OpenSSH



- Open Source DBMS (MySQL, PostgreSQL, and Firebird) used by 64% of developers and database administrators of those who use FLOSS – February 2005 [Evans Data Corp.]
- OpenOffice.org in 2004 had 14% large enterprise office systems market (MS 95% overall) [CSC]

Reliability

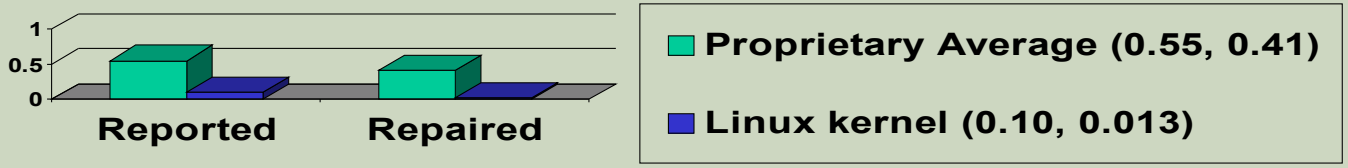
- **Fuzz studies found FLOSS applications significantly more reliable [U Wisconsin]**



- **GNU/Linux vs. Windows NT 10 mo study [ZDNet]**
  - **NT crashed every 6 weeks; both GNU/Linuxes, never**
- **IIS web servers >2x downtime vs. Apache [Syscontrol AG]**
- **Survey of 6MLOC: OSS "maintainability index" equal & sometimes better vs. closed [Samoladas in CACM, Oct 2004]**
- **FLOSS: More modular [MacCormack, Harvard Bus. School]**

Reliability (2)

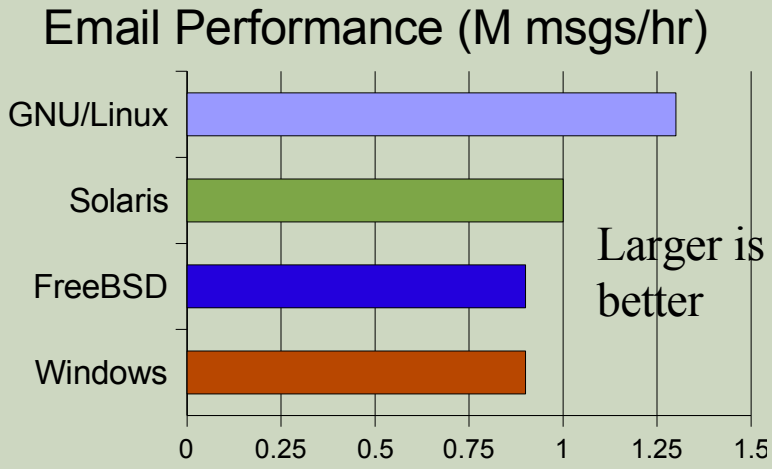
- **Automated defect detection analysis:**
  - **Linux kernel: of 5.7MSLOC, only 985 detected (>5000 expected, 80% fewer) [Coverity]**
  - **MySQL: 0.09 defects/KSLOC vs. 0.57 average defects/KSLOC avg. 200 proprietary [Reasoning]**
  - **Linux kernel TCP/IP had smaller defect density [Reasoning]**

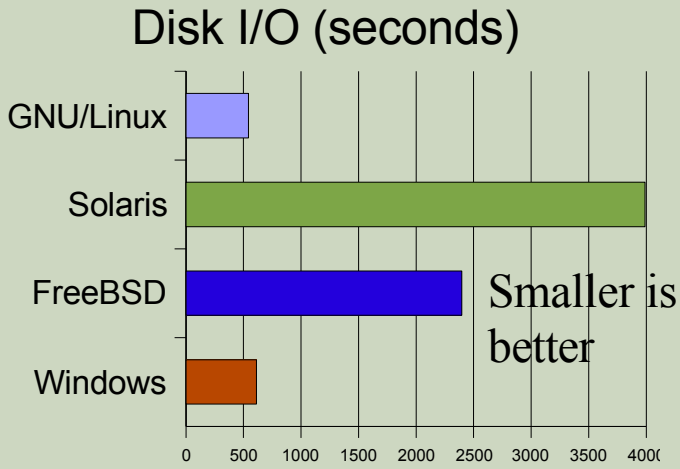


Performance: GNU/Linux

Performance varies widely by circumstance!

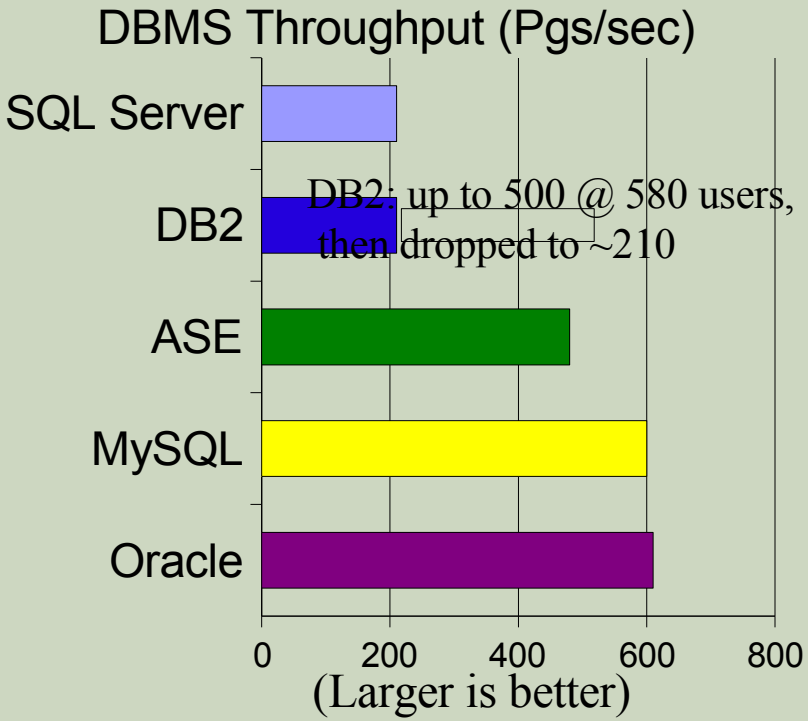
- **GNU/Linux with Samba faster fileserving at Windows' own file protocols [PC Magazine]**
  - **Nov 2001, top end, 130MB/sec vs. 78MB/sec**
  - **April 2002, performance 2x; 4x many clients**
- **GNU/Linux fastest (untuned systems) [Sys Admin]**

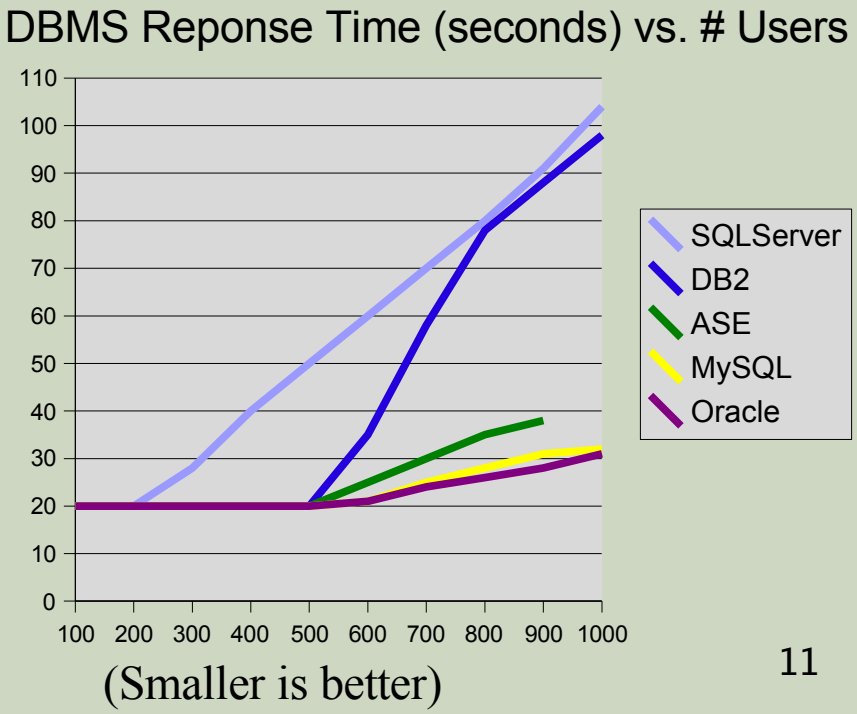




Performance: DBMSs

- **eWeek Labs/PC Labs 2002 DBMS evaluation**
  - **Unusual; most DBMS licenses forbid publication**
  - **MySQL (FLOSS) did *very* well**





Scalability

- GNU/Linux and NetBSD support more hardware platforms & performance ranges than any other
  - PC hardware, PDAs, mainframes, clusters, supercomputers
- 78% of supercomputers run GNU/Linux as of Nov. 2005, up from 60% March 2005 [Top500.org]
- FLOSS can develop large software systems
  - Red Hat Linux 7.1 had 30million SLOC
  - Represents approximately 8,000 person-years
  - To re-develop proprietary, $1 Billion USD (a “Gigabuck”) [Wheeler]

Security

- **J.S. Wurzler hacker insurance costs 5-15% more for Windows than for Unix or Linux**
- **Windows websites disproportionately vulnerable**

Category	Proprietary	FLOSS
Defaced	**66% (Windows)**	**17% (GNU/Linux)**
Deployed Systems	**49.6% (Windows)**	**29.6% (GNU/Linux)**
Deployed websites (by name)	**24.81% (IIS)**	**66.75% (Apache)**

- **Bugtraq vulnerability 99-00: Smallest is OpenBSD, Windows largest (**Don't quintuple-count!**)**
- **Worst vulnerabilities (takeover): Apache 0, IIS 8 (Jun98-Jun01)**
- **Browser "unsafe" days in 2004: 98% Internet Explorer, 15% Mozilla/Firefox**

Security (2)

- Unpatched networked systems: 3 months Linux, hours Windows (variance minutes ... months) [Honeynet.org, Dec 2004]
  - Windows SP2 believed to be better than previous versions of Windows
- 50% Windows vulnerabilities are critical, vs. 10% in Red Hat [Nicholas Petreley, Oct 2004]
- Viruses primarily Windows phenomenon
  - 60,000 Windows, 40 Macintosh, 5 for commercial Unix versions, 40 for Linux
- 91% broadband users have spyware on their home computers (proprietary OS) [National Cyber Security Alliance, May 2003] vs. ~0% on FLOSS

Security (3)

- FLOSS systems scored better on security [Payne, Information Systems Journal 2002]

	Debian	Solaris	OpenBSD
Number of Features:	15	11	18
Features score:	6.42	5.92	7.03
Number of Vulnerabilities:	12	21	5
Vulnerabilities score:	7.72	7.74	4.19
Final Score (larger better):	-1	-3.5	10.2

- Survey of 6,344 software development managers April 2005 favored FLOSS [BZ Research]

	MS Windows Server	Linux	Sun Solaris
Very insecure or Insecure:	58%	6%	13%
Secure or very secure:	38%	74%	66%

What's more Secure?	OSS/FS	Proprietary
Desktop/Client OS:	58%	6%
Web Servers:	43%	14%
Server OS:	38%	22%
Components/Libs:	34%	18%
Database Servers:	21%	34%

Total Cost of Ownership (TCO): Background

- TCO multifaceted; for software-based system: [CSC]
  - Direct software costs (purchase, maintenance, support)
  - Indirect software costs (license admin, audit)
  - Hardware (purchase/upgrade, maintenance, dispose)
  - Staffing (project management, systems engineering, administration (e.g., purchasing), systems admin)
  - Support (install, troubleshoot, casual learning, training)
  - Downtime
- TCO sensitive to circumstances
  - Helpful for single decision, hard to generalize
  - *Anything* has a lower TCO for *some* circumstance
  - Architecture matters!: Independent clients, X-terms, stateless, cluster, etc. *May be best deployed differently*
- Really "Total Cost to Lease," esp. for proprietary

TCO: General FLOSS

- FLOSS usually costs less to acquire than proprietary
  - E.G., Web server, Windows $3610 vs. $156
- Some other factors also tend to be lower
  - Lower upgrade costs, can use cheaper hardware
  - Avoids license management & litigation
  - Downtime less: more modular, remove unneeded [CSC]
- Maintenance/Support: Varies, can be competed
- Cybersource: TCO 24%-34% less w/FLOSS
- InfoWorld Survey of CTOs:
  - 60% CTOs: >$50K/yr savings
  - 32% CTOs: > $250K/yr savings (inc. above)
- Survey of companies > $5M revenue[InternetWeek/InformationWeek]
  - 39%: FLOSS costs 25% to 50% less
  - 27%: FLOSS costs 50% to 75% less

TCO: Specific Examples

- **Measured Web server TCO of GNU/Linux is 40% (<1/2) of Windows' and 14% of Solaris' [RFG]**
- **Amazon.com: $17M savings in 1Q via Linux**
- **UK Gov't Becta* 3yr study: FLOSS savings significant in primary & secondary schools**
  - **Secondaries reduce IT overheads by 24% (inc. software, hardware, and support costs)**
  - **Primary schools cut computer costs by nearly half, primarily from support but also hardware**
- **Willamette U. Library $41K vs. $100-150K using networked X terminals [Murphy]**
- **Netproject: Desktop Linux 35% (save 65%!) of Windows**
- **Largo, FL: $1M/yr savings thin clients**

*Becta: British Educational Communications and Technology Association

Non-Quantitative

- To many, non-quantitative advantages of FLOSS are more important, e.g.:
  - Social/ethical/moral reasons
  - Avoids risks of single source solutions/lock-in
    - Create reversible decision: can switch/self-support if price jacked up, maliciously changes interface, drops support, needs change (can get data), …
  - (Can) avoid security risks of monocultures
  - Supports domestic IT infrastructure
  - Long-term data retention (format not secret)
  - Many believe it encourages innovation
  - Avoids license management and litigation
  - Greater flexibility
    - Can change software (inc. via hiring) to meet needs

Conclusions

- FLOSS in many cases have measurable advantages over proprietary competition
- Consider using FLOSS software when acquiring
- Don't disadvantage FLOSS in policy
  - Be wary of vendor lock-in
  - Prefer open standards (publicly held, multi-vendor support, don’t require patents)
  - Beware of “vendor pays” assumptions (CC)
  - Software patents justified?
- For more detailed information, see http://www.dwheeler.com/oss_fs_why.html

Backup Slides

- Introduction to FLOSS
  - Basics, history, OSS vs. FS, licenses, development model
- Unnecessary fears
- Acronyms
- Interesting sites/documents

Basics of FLOSS: Free-Libre / Open Source Software (OSS)

- **Free-Libre / Open Source Software (FLOSS) programs have licenses giving users the freedom:**
  - **to run the program for any purpose,**
  - **to study and modify the program, and**
  - **to freely redistribute copies of either the original or modified program (without royalties, etc.)**
- ***Not* non-commercial, *not* necessarily free-of-charge**
  - **Often supported via commercial companies**
- **Synonyms: Libre software, FLOS, OSS/FS**
- **Antonyms: proprietary software, closed software**

History of FLOSS

- 1950s, 1960s: Software freely distributed
- ~1970s: Rise of proprietary software
- 1984: Richard M. Stallman establishes "Free Software Foundation", creates "General Public License" (GPL)
- 1990s: Increasing Internet availability enables developer coordination
- 1997: Eric Raymond's "Cathedral & the Bazaar" explains new approaches; term "Open Source Software" coined

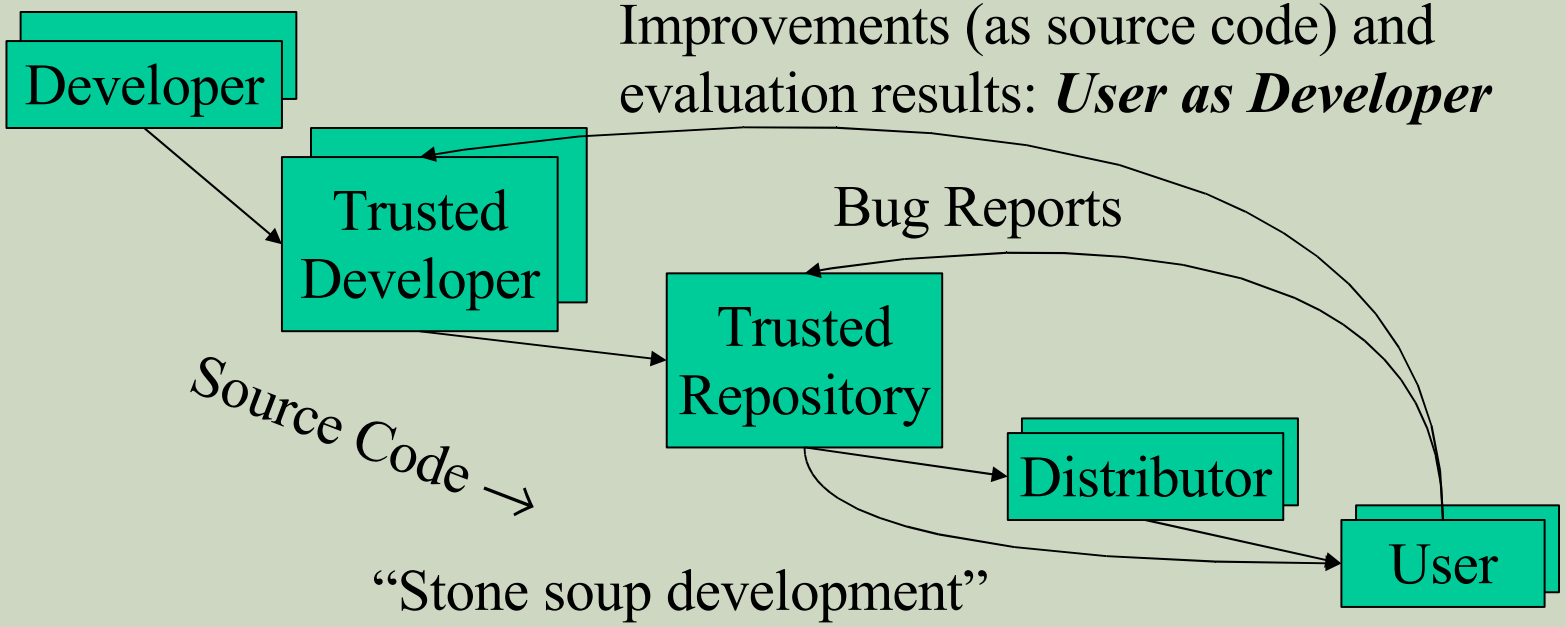
"Open Source Software" vs. "Free Software"

- First named "Free Software" by Stallman
  - Free as in Freedom
  - Officially defined by "Free Software Definition"
  - *Not* necessarily zero price; confused many
- New term coined: "Open Source Software"
  - Officially defined by the "Open Source Definition" (a long 9-point list)
  - Practically all OSS software is also FS
- Terms sometimes indicate motivations
  - FS: emphasize ethical/social issues
  - OSS: technical superiority/flexibility
  - OSS often used due to "zero price" confusion

Major FLOSS Licenses

- Many licenses, but 4 dominate
- BSD-new & MIT license: anything but sue
  - Can incorporate code into proprietary software
  - Financial incentive to use, but *not* aid project
- General Public License (GPL): "Copyleft"
  - If distribute, *must* distribute source code or provide written offer to do so
  - Cannot link (embed) into proprietary software
- Lesser/Library GPL - a compromise
  - Must distribute source code/written offer, but only of component itself
  - *Can* link into proprietary software
- Public domain is FLOSS, but rare

FLOSS Development Model



- FLOSS users typically use software without paying licensing fees
- FLOSS users typically pay for training & support (competed)
- FLOSS users are responsible for developing new improvements & any evaluations that they need; often cooperate/pay others to do so

Unnecessary Fears

- **Proprietary software always better supported? No.**
  - **Non-traditional support (mailing lists, etc.)**
  - **Pay for traditional support, and can compete it**
- **Proprietary more legal rights? No.**
  - **Who do you sue? Nobody, in either case**
- **FLOSS economically viable? Yes.**
  - **Many business models**
  - **Customers can band together**

Unnecessary Fears

- **Will programmers starve? No.**
  - **Estimated 95% software not developed for sale**
  - **Companies hire programmers to make changes for themselves**
  - **Widespread use of FLOSS moves software development into a service (not product) model**
- **FLOSS compatible with capitalism? Yes.**
  - **FLOSS development involves trade: code for code**
  - **FLOSS business often based on payment for support or commoditizing complements of products**
- **FLOSS mean no competition? No.**
  - **KDE vs. GNOME, emacs vs. vim**

Unnecessary Fears

- **Will FLOSS destroy intellectual property? No.**
  - **Usually, complaint is about GPL**
  - **GPL trades you the right to freely incorporate their code into your software in exchange for the right to freely incorporate your code [which incorporates their code] into theirs**
  - **Intellectual property traded for other intellectual property**
  - **Microsoft sells GPL'ed software, sponsored several FLOSS projects**

Unnecessary Fears

- Viewing and changing source code valuable for non-programmers? Surprisingly, yes.
  - "Would you buy a car with the hood welded shut? If not, what do you know about modern … engine technology?" [Bob Young]
  - Consumers demand this so they can have control over their product support, instead of dealers
- Anti-Microsoft campaign? No, not by all.
  - Jun02, 831 projects use Visual Basic; 8867 projects work on Windows [SourceForge]
  - Microsoft has been repeatedly asked to join community
  - Microsoft long used, and now develops FLOSS
  - Microsoft has sold GPL'ed software

Acronyms

- COTS: Commercial Off-the-Shelf (either proprietary or OSS)
- DoD: Department of Defense
- HP: Hewlitt-Packard Corporation
- JTA: Joint Technical Architecture (list of standards for the DoD); renamed to DISR
- OSDL: Open Source Development Labs
- OSS: Open Source Software
- RFP: Request for Proposal
- RH: Red Hat, Inc.
- U.S.: United States

Interesting Documents/Sites

- **“Why OSS/FS? Look at the Numbers!” (larger paper) http://www.dwheeler.com/oss_fs_why.html**
- **“Use of Free and Open Source Software in the US Dept. of Defense” (MITRE, sponsored by DISA)**
- **President's Information Technology Advisory Committee (PITAC) -- Panel on Open Source Software for High End Computing, October 2000**
- **“Open Source Software (OSS) in the DoD,” DoD memo signed by John P. Stenbit (DoD CIO), May 28, 2003**
- **Center of Open Source and Government (EgovOS) http://www.egovos.org/**
- **OpenSector.org http://opensector.org**
- **Open Source and Industry Alliance http://www.osaia.org**
- **Open Source Initiative http://www.opensource.org**
- **Free Software Foundation http://www.fsf.org**
- **OSS/FS References http://www.dwheeler.com/oss_fs_refs.html**

Presentation License

- This presentation (C) Copyright 2005 David A. Wheeler. You may distribute unmodified and modified derivative copies; pick from:
  - Creative Commons Attribution-ShareAlike License version 2.5 or greater
  - GNU GPL version 2 or greater
  - GNU Free Documentation License (GFDL) version 1.2 or greater
- Please include a reference to http://www.dwheeler.com/numbers and http://www.dwheeler.com/oss_fs_why.html

Why Free-Libre / Open Source Software (FLOSS)? Look at the Numbers!

David A. Wheeler

November 15, 2005

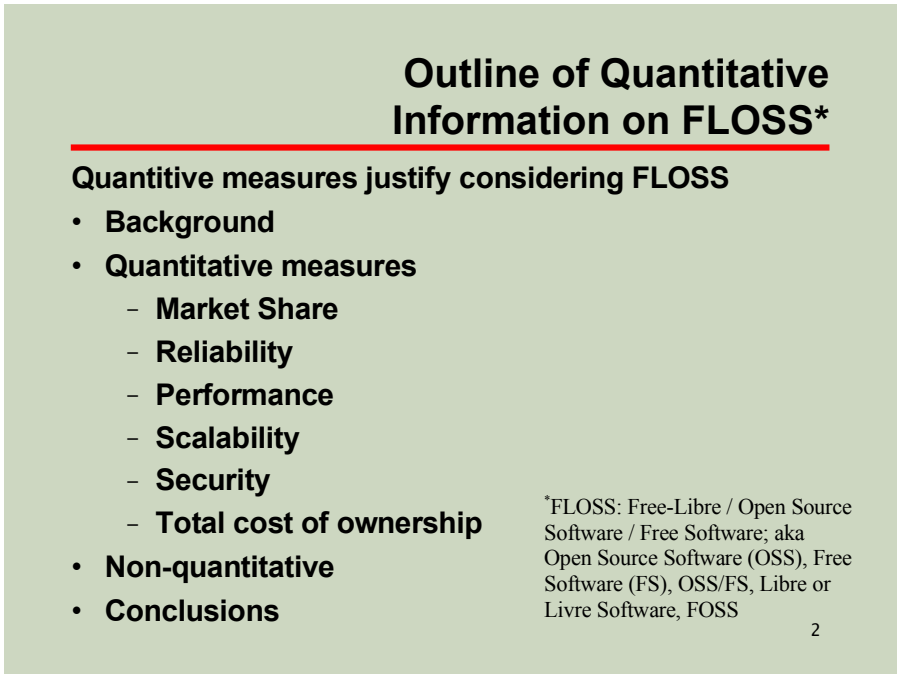
http://www.dwheeler.com/numbers
http://www.dwheeler.com/oss_fs_why.html

This presentation contains the views of the author and does not necessarily indicate endorsement by IDA, the U.S. government, or the U.S. DoD.

1

This is an annotated presentation titled "Why Free-Libre / Open Source Software (FLOSS)? Look at the Numbers!" You can get the current copy of this presentation at "http://www.dwheeler.com/numbers". This presentation is a summary of a much larger paper with a similar title, "Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!" That larger paper is available at http://www.dwheeler.com/oss_fs_why.html.

This presentation contains the views of the author and does not necessarily indicate endorsement by IDA, the U.S. government, or the U.S. DoD.



As you can guess from the title, this is a survey of many quantitative studies involving Free-Libre / Open Source Software (FLOSS). Broadly speaking, FLOSS programs have licenses giving users the freedom to run the program for any purpose, to study and modify the program, and to freely redistribute copies of either the original or modified program (without royalties, etc.).

The purpose of this survey is to convince you to consider using FLOSS when you're looking for software. Some sites provide a few anecdotes on why you should use FLOSS, but I (and many others) find such anecdotes unconvincing. Instead, this presentation emphasizes quantitative measures (such as experiments and market studies) to justify why using FLOSS products is in many circumstances a reasonable or superior approach. I should note that while I find much to like about FLOSS, I'm not a rabid advocate; I use both proprietary and FLOSS products myself. Vendors of proprietary products often work hard to find numbers to support their claims; this material provides a useful antidote of hard figures to aid in comparing proprietary products to FLOSS. Hopefully it's enough to convince you to consider FLOSS alternatives when you are acquiring software.

This presentation, as well as the paper it summarizes, shows a set of quantitative data, grouped into the following categories: market share, reliability, performance, scalability, security, and total cost of ownership. It briefly notes some of the non-quantitative reasons people choose FLOSS, and closes with conclusions.

FLOSS is also known as open source software (OSS), Free Software (FS), Open Source Software / Free Software (OSS/FS), or Free/Open Source Software (FOSS). Some people who use the term "Free Software" (where "free" means "freedom" not "no cost") have different motivations than those who use the term "open source software," but those distinctions aren't the focus of this talk. Many FLOSS projects are commercial, so do not confuse "FLOSS" with "non-commercial."

Background

- **In 2000, many claims about FLOSS, yet their advocates gave little evidence**
- **Investigated & found there *was* evidence**
- **Collection now widely-referenced**
  - **California Performance Review, 2004**
  - **Google "open source software" #5**
- **Challenges:**
  - **Vendor-funded studies (conflict of interest)**
  - **Some proprietary licenses forbid speech**
- **Numbers can't prove "always better"**

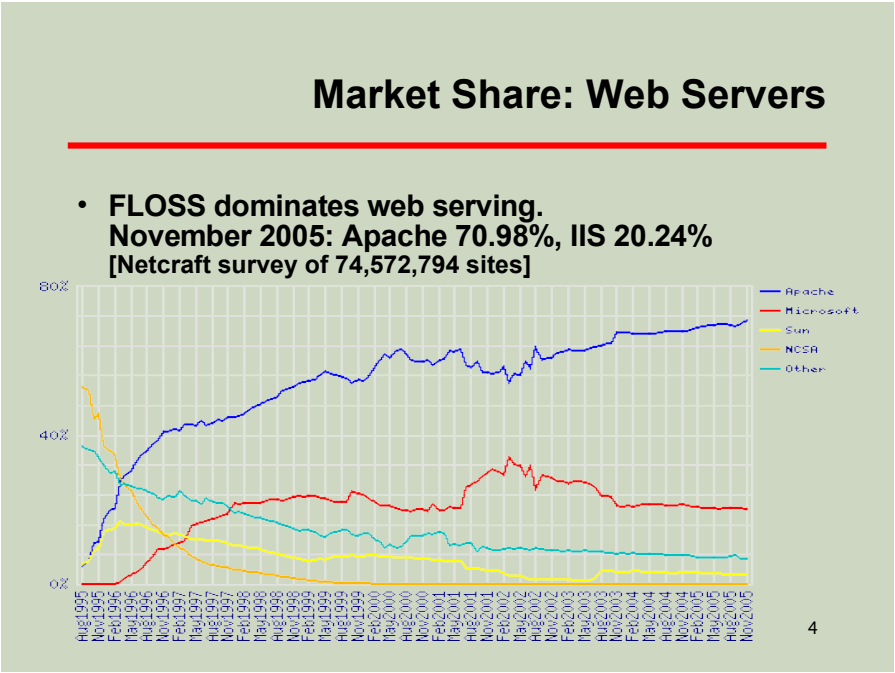
3

Before 2000, many were claiming that FLOSS had a large number of advantages, including lower total cost of ownership (due to licensing differences), reliability (due to mass peer review), and so on.. Yet their advocates provided little evidence for their claims. I investigated and found that there *was* evidence that at least some FLOSS programs did very well by various measures, yet there was no single place where this data could be found. So I ended up collecting it, and that set has become substantial.

This work is now widely referenced. The 2004 California Performance Review, for example, cited my work, and a Google search on "open source software" places my paper at number 5 (a very high placement). I say this as evidence that many others have found this collection of information useful.

There are challenges to any work like this. There are a number of vendor-funded studies. Vendor-sponsored studies are often rigged (no matter who the vendor is) to make the vendor look good instead of being fair comparisons. There is a fundamental conflict-of-interest in such studies, and there are many ways to rig a study. Also, some proprietary software licenses forbid public benchmarks or critiques unless the vendor approves it, and some lawmakers have allowed this chilling of free speech. As a result, some data is harder to get.

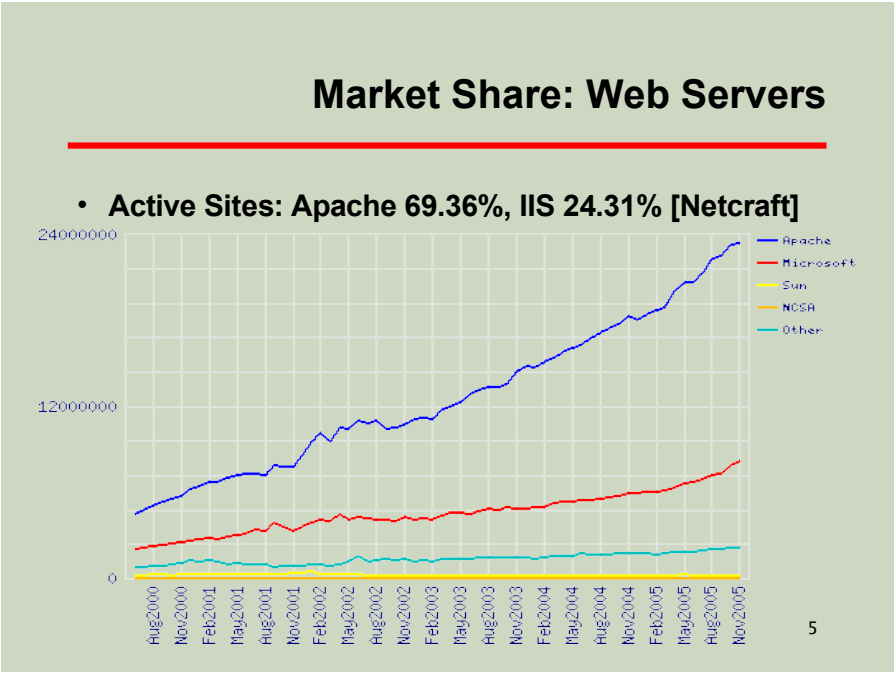
Numbers cannot prove "FLOSS is always better," nor is that my goal. My goal is to simply convince you that FLOSS is worth investigating for your particular circumstance, by using a mountain of evidence.



We'll start first by looking at the data on market share. Many people think that a product is only a winner if it has significant market share. This is lemming-like, but there's some rationale for this: products with big market shares get applications, trained users, and momentum that reduces future risk. Herd animals are protected by the herd; the same can be said of using products with significant market share. As we'll see, there are number of markets where FLOSS has a significant presence. In some other cases, the number is smaller but the trend is increasing; trends in some ways are more important, because they help us predict what is likely to be dominant in the future.

Here we see market share (as measured by domain names) of various web servers. For as long as such markets have been measured, FLOSS programs have *always* dominated this market. Early on the original NCSA web server dominated until Apache came; Apache has dominated the web server market ever since. Note that it's not even a close contest; Apache simply dominates the other market players, with nearly 70% of this market, and the trends suggest that this is unlikely to change in the near future.

This data is provided by Netcraft, http://www.netcraft.com. Other surveys, independent of Netcraft, show the same thing: Apache is the dominant web server, and it is OSS/FS.

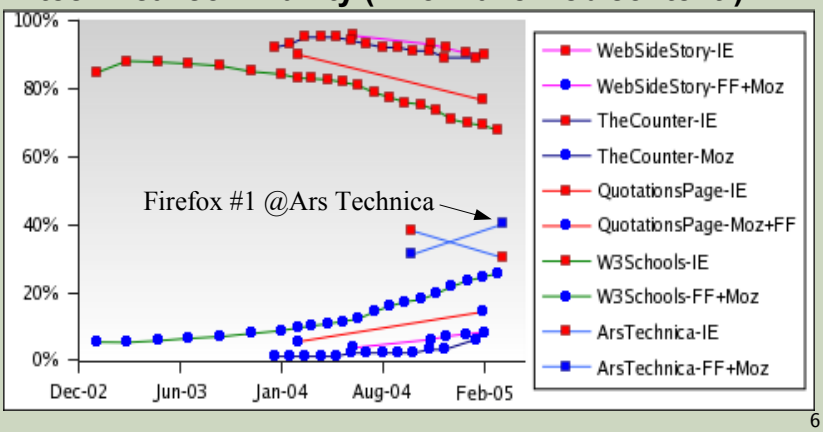


Here's more Netcraft data, which measures web server market share in a slightly different way.

One problem with measuring web server market share is that some domain names have a website, but the website is not being actively used. Such websites are simply placeholders (e.g., speculative domain name purchasers or planned websites that have not yet appeared). More recently Netcraft has been separately tracking "active sites," that is, sites that are not simply placeholders. Apache does quite well here too; Apache has most of the market by this measure. This chart also shows the raw number of servers (instead of percentages). This shows even more remarkably how many web sites are run by the FLOSS program Apache, and how their numbers are rapidly growing.

Web Browsers: Growing Fast

- **Mozilla/Firefox use growing, esp. among web/ technical community (who make web content!)**



6

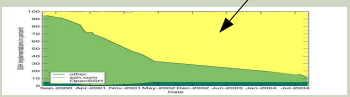
What about Web browsers? Here, Internet Explorer has dominated, but it is now losing market share. This trend is important; we will all be living the future, not the present, and trends help us understand what that future will be like.

Different surveys produce different results, because they use different sample populations, but all agree on one point: Mozilla/Firefox is gaining market share, and Internet Explorer is losing it. This is particularly obvious at blogging sites (heavy users of web browsers) and technical people (who will be developing your next web servers). These leading-edge people have switched even more dramatically, suggesting that this may simply be the leading edge of a larger trend. For example, by February 2005, more clients were using Firefox than Internet Explorer at the technical site "Ars Technica."

Other Market Share Examples

- **GNU/Linux #2 server OS sold 99, 00, 01 (24%, 27%, 25%)**
- **DNS: bind supports 95% of reverse-lookups [Manning]**
- **PHP #1 server-side scripting language [Netcraft]**
- **Sendmail #1 Email server [Bernstein]**
  - **Sendmail 42%, Microsoft Exchange 18%**
- **OpenSSH #1 SSH (87.9% Sep04) ~5% Summer 2000, 50% November 2001 [scanssh]**
- **Open Source DBMS (MySQL, PostgreSQL, and Firebird) used by 64% of developers and database administrators of those who use FLOSS – February 2005 [Evans Data Corp.]**
- **OpenOffice.org in 2004 had 14% large enterprise office systems market (MS 95% overall) [CSC]**

OpenSSH



7

There are many other quantitative studies that discuss market share:

• GNU/Linux has been the #2 server operating system, in terms of volume, in at least 1999, 2000, and 2001.

• The Domain Name Service (DNS) is critical to having a working Internet. Manning realized that although "ordinary" queries would be hard to survey, "reverse-lookup" queries are relatively easy to survey – and he found that 95% of all such queries are serviced by bind, an FLOSS program.

• Sendmail is still the #1 program used to transfer email; it's FLOSS.

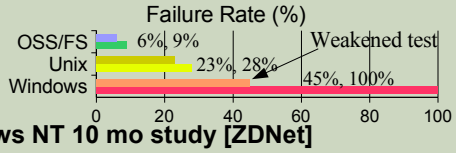
• OpenSSH is an important program used to securely control computers remotely. In 2000 the FLOSS version was rarely used compared to proprietary implementations; by 2004 there were almost no proprietary implementations left in use.

• FLOSS database management systems (DBMSs) have become wildly popular

OpenOffice.org has become a widely-used office suite, especially since it works on Windows, GNU/Linux, Unix, and MacOS. While Microsoft Office still has far more total users, OpenOffice.org had 14% of the large enterprise office systems market in 2004.

Reliability

- **Fuzz studies found FLOSS applications significantly more reliable [U Wisconsin]**



- **GNU/Linux vs. Windows NT 10 mo study [ZDNet]**
  - **NT crashed every 6 weeks; both GNU/Linuxes, never**
- **IIS web servers >2x downtime vs. Apache [Syscontrol AG]**
- **Survey of 6MLOC: OSS "maintainability index" equal & sometimes better vs. closed [Samoladas in CACM, Oct 2004]**
- **FLOSS: More modular [MacCormack, Harvard Bus. School]**

8

We now come to studies about reliability. Study after study shows that FLOSS can be extremely reliable.

One of the broadest reliability studies that covered FLOSS were the "fuzz studies" of the University of Wisconsin. Here they ran many different programs, sending the programs large sets of random values. A program would "pass" this test if it didn't crash or "hang" (stay stuck for a long time) – regardless of whether it got the "right" answer. This simple approach can test *any* program, in a fair way, and since "correct" answers aren't needed a massive number of test cases can be used. Thus, unlike most studies, this study broadly compares a large sample of FLOSS programs against a large sample of proprietary programs (instead of just individual comparisons). In theory, no program should fail such a trivial test, but that's not how things worked out. Of a set of Unix programs, 28% failed; when they tried again 5 years later, 23% failed (including some of the same failures).

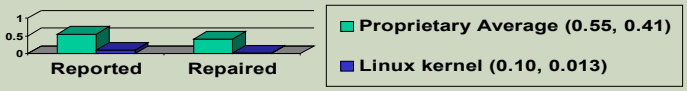
In their second series of tests they also tested FLOSS programs, and they were astonished. Compared to Unix's 23%, only 9% of the FLOSS programs failed, and when they used the GNU subset (which were more mature) only 6% failed. In contrast, the Windows results were disturbing: *every* Windows program failed. When they *changed the test conditions* to make a weaker test for just the Windows programs, the Windows programs were still far more likely to fail (with this easier test) than their competition (even though they were measured in harsher way).

ZDNet ran a test of two GNU/Linux systems against Windows NT, with *exactly* the same requests for 10 months. NT crashed every 6 weeks; Linux, never. This also explains why such studies are rare – they take a long time to run.

Syscontrol AG found that IIS servers were down twice as much as Apache. Various code studies have found FLOSS have better code measures, and better code is correlated with better reliability.

Reliability (2)

- **Automated defect detection analysis:**
  - **Linux kernel: of 5.7MSLOC, only 985 detected (>5000 expected, 80% fewer) [Coverity]**
  - **MySQL: 0.09 defects/KSLOC vs. 0.57 average defects/KSLOC avg. 200 proprietary [Reasoning]**
  - **Linux kernel TCP/IP had smaller defect density [Reasoning]**



9

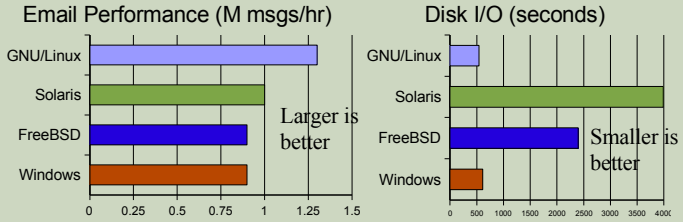
There are a number of tools that attempt to do automated defect detection analysis. They can also be used to estimate the number of probable defects in code, and then those values can be compared.

The Linux kernel and MySQL have been analyzed by such tools, and found that the density of errors (the likelihood that a given line has a defect) is far lower in these FLOSS programs than for typical proprietary programs.

Performance: GNU/Linux

Performance varies widely by circumstance!

- **GNU/Linux with Samba faster fileserving at Windows' own file protocols [PC Magazine]**
  - **Nov 2001, top end, 130MB/sec vs. 78MB/sec**
  - **April 2002, performance 2x; 4x many clients**
- **GNU/Linux fastest (untuned systems) [Sys Admin]**



10

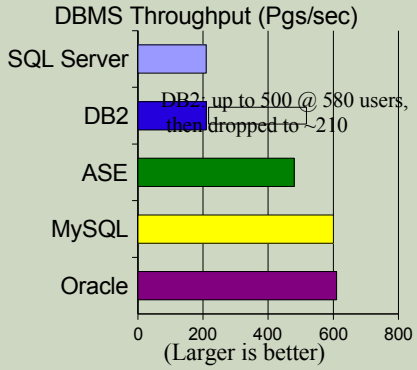
Performance varies widely by circumstance, so any such measures need to be considered skeptically. You must measure performance for your particular circumstance to determine performance in that case, where that is important. Still, available performance measures suggest that FLOSS can have good performance.

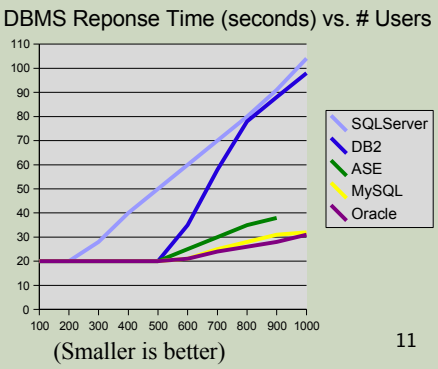
PC Magazine found that (FLOSS) GNU/Linux served Windows' file serving protocols faster than Windows did in 2001 and 2002.

Sys Admin measured several systems (without "tuning" them, that is, without working hard to configure them for maximum performance). The GNU/Linux systems did the best at both email performance and disk input/output.

Performance: DBMSs

- **eWeek Labs/PC Labs 2002 DBMS evaluation**
  - **Unusual; most DBMS licenses forbid publication**
  - **MySQL (FLOSS) did *very* well**





11

In 2002 something extraordinary happened – there was a published DBMS evaluation. This is unusual because the licenses of many companies (such as Oracle) forbid them, but eWeek Labs/PC Labs challenged companies to sue.

Here Oracle does slightly better than FLOSS MySQL, but only slightly, and Oracle costs orders of magnitude more money than MySQL. MySQL handily beats its other rivals.

Scalability

- GNU/Linux and NetBSD support more hardware platforms & performance ranges than any other
  - PC hardware, PDAs, mainframes, clusters, supercomputers
- 78% of supercomputers run GNU/Linux as of Nov. 2005, up from 60% March 2005 [Top500.org]
- FLOSS can develop large software systems
  - Red Hat Linux 7.1 had 30million SLOC
  - Represents approximately 8,000 person-years
  - To re-develop proprietary, $1 Billion USD (a "Gigabuck") [Wheeler]

12

Another kind of measure is "scalability," but there are different meanings for the term.

If the issue is what has the widest computing platform range (from small computer to supercomputer) then FLOSS has no competition. GNU/Linux and NetBSD, both FLOSS, support more hardware platforms and performance ranges than any other.

Most of the supercomputers of world run the FLOSS operating system GNU/Linux. In March 2005 it was 60%, and by November 2005 it was 78%; in both cases the fastest computer in the world ran the FLOSS GNU/Linux. Clearly FLOSS is not limited to being a toy.

It's also clear that FLOSS can develop large systems. I found that Red Hat Linux 7.1, now a very old distribution, had 30 million physical source lines of code (SLOC). It represented approximately 8,000 person-years and $1 Billion U.S. dollars (if it had to be redeveloped in a traditional proprietary manner). This "billion" is one thousand million dollars, or as I term it, a "Gigabuck."

Security

- **J.S. Wurzler hacker insurance costs 5-15% more for Windows than for Unix or Linux**
- **Windows websites disproportionately vulnerable**

Category	Proprietary	FLOSS
Defaced	66% (Windows)	17% (GNU/Linux)
Deployed Systems	49.6% (Windows)	29.6% (GNU/Linux)
Deployed websites (by name)	24.81% (IIS)	66.75% (Apache)

- **Bugtraq vulnerability 99-00: Smallest is OpenBSD, Windows largest (**Don't quintuple-count!**)**
- **Worst vulnerabilities (takeover): Apache 0, IIS 8 (Jun98-Jun01)**
- **Browser "unsafe" days in 2004: 98% Internet Explorer, 15% Mozilla/Firefox**

13

Security is obviously a very important topic, and one that is very difficult to measure. Still, there are measures that suggest that FLOSS can be quite secure compared to its competition.

Insurance company J.S. Wurzler sells "hacker insurance"; based on its actuarial data, it gives Unix and Linux systems a discount since they're less risky to insure.

Windows sites are disproportionately vulnerable to attack.

A count of Bugtraq vulnerabilities in 1999-2000 found the smallest to be the FLOSS OpenBSD, while Windows had the most. Warning: several reporters have independently made the same mistake, quintuple-counting the same vulnerability.

I studied the time period June 1998 to June 2001 for takeover vulnerabilities, and found that Apache did very well (no such vulnerabilities!) while Microsoft's competing IIS did poorly by comparison.

One study of browser vulnerabilities in 2004 counted the number of days where a vulnerability to the worst kind of attack (execution of program from a remote attacker) was publicly known, yet there was no patch available to fix it. Microsoft's Internet Explorer did absurdly poorly – it was unsafe 98% of the time. There were only 7 days (in October) where Internet Explorer could be safely used. Mozilla/Firefox had this problem only 15% of the days, even better than the 17% by the proprietary program Opera.

Security (2)

- **Unpatched networked systems: 3 months Linux, hours Windows (variance minutes ... months) [Honeynet.org, Dec 2004]**
  - **Windows SP2 believed to be better than previous versions of Windows**
- **50% Windows vulnerabilities are critical, vs. 10% in Red Hat [Nicholas Petreley, Oct 2004]**
- **Viruses primarily Windows phenomenon**
  - **60,000 Windows, 40 Macintosh, 5 for commercial Unix versions, 40 for Linux**
- **91% broadband users have spyware on their home computers (proprietary OS) [National Cyber Security Alliance, May 2003] vs. ~0% on FLOSS**

14

Honeynet.org did a study and found that while Linux systems tended to take 3 months to crack, in general Windows systems could only last for "hours" on the Internet before they were taken over.

Nicholas Petreley studied vulnerabilities, and found that the Windows systems' vulnerabilities tended to be more serious: 50% of the Windows vulnerabilities were critical, compared to 10% for Red Hat.

And of couse, viruses and spyware are essentially only seen on proprietary systems.

Security (3)

- **FLOSS systems scored better on security [Payne, Information Systems Journal 2002]**

	Debian	Solaris	OpenBSD
Number of Features:	15	11	18
Features score:	6.42	5.92	7.03
Number of Vulnerabilities:	12	21	5
Vulnerabilities score:	7.72	7.74	4.19
Final Score (larger better):	-1	-3.5	10.2

- **Survey of 6,344 software development managers April 2005 favored FLOSS [BZ Research]**

	MS Windows Server	Linux	Sun Solaris
Very insecure or Insecure:	58%	6%	13%
Secure or very secure:	38%	74%	66%

What's more Secure?	OSS/FS	Proprietary
Desktop/Client OS:	58%	6%
Web Servers:	43%	14%
Server OS:	38%	22%
Components/Libs:	34%	18%
Database Servers:	21%	34%

15

Payne did a study of Debian, Solaris, and OpenBSD. At the time Solaris was proprietary; Debian and OpenBSD were FLOSS. By Paynes' measures, OpenBSD did by far the best; Solaris, the worst.

BZ research did a survey of 6,344 software development managers, and found that in the managers' experience, the FLOSS programs tended to be better (with one exception, database servers, which was a somewhat close 34% to 21%).

Total Cost of Ownership (TCO): Background

- **TCO multifaceted; for software-based system: [CSC]**
  - **Direct software costs (purchase, maintenance, support)**
  - **Indirect software costs (license admin, audit)**
  - **Hardware (purchase/upgrade, maintenance, dispose)**
  - **Staffing (project management, systems engineering, administration (e.g., purchasing), systems admin)**
  - **Support (install, troubleshoot, casual learning, training)**
  - **Downtime**
- **TCO sensitive to circumstances**
  - **Helpful for single decision, hard to generalize**
  - ***Anything* has a lower TCO for *some* circumstance**
  - **Architecture matters!: Independent clients, X-terms, stateless, cluster, etc. *May be best deployed differently***
- **Really "Total Cost to Lease," esp. for proprietary**

16

"Total cost of ownership" (TCO) is in theory a simple measure – it's just the cost of a solution over a period of time. TCO depends on many values including factors like staffing, support, and downtime.

TCO is *extremely* sensitive to specific circumstances. As a result, TCO values can be extremely helpful for making a specific decision (for that circumstance), but can be hard to generalize to other circumstances. Indeed, just about any approach has a lower TCO for *some* circumstance.

In particular, note TCO is sensitive to the proposed system architecture. FLOSS systems can often be deployed in different ways, and may be best deployed in one of those alternative architectures. Approaches such as using "X terminals," "stateless clients," and "clusters" should be considered where appropriate.

The term TCO is common but misleading for most software, especially for proprietary software, because software users usually don't own the software they use and thus don't have the rights of ownership. It might be more accurate to say that proprietary software users often "lease" or "rent" the software. Unless you arrange to have a software program's copyright transferred to you, you do not actually own the software – you only own a license to run the software in certain limited ways. Unlike owning a physical product, with nearly all proprietary software you cannot modify the software to suit your needs, nor examine the product to understand what it does, nor publish benchmarks or reviews without vendor approval, and often cannot (sub)lease, resell, or redistribute the product. In contrast, you can execute these ownership-like rights with FLOSS. These kinds of limits make proprietary software use more like a lease or a rent.

TCO: General FLOSS

- FLOSS usually costs less to acquire than proprietary
  - E.G., Web server, Windows $3610 vs. $156
- Some other factors also tend to be lower
  - Lower upgrade costs, can use cheaper hardware
  - Avoids license management & litigation
  - Downtime less: more modular, remove unneeded [CSC]
- Maintenance/Support: Varies, can be competed
- Cybersource: TCO 24%-34% less w/FLOSS
- InfoWorld Survey of CTOs:
  - 60% CTOs: >$50K/yr savings
  - 32% CTOs: > $250K/yr savings (inc. above)
- Survey of companies > $5M revenue[InternetWeek/InformationWeek]
  - 39%: FLOSS costs 25% to 50% less
  - 27%: FLOSS costs 50% to 75% less

17

Some factors do often drive FLOSS costs to be lower. FLOSS is almost universally cheaper to acquire initially and to upgrade. Hardware requirements are often lower. One often-forgotten issue is that license management and litigation is far simpler for FLOSS. CSC reports that downtime tends to be less; since FLOSS is more modular, it's easy to remove unneeded components, and a component that isn't there can't fail.

Maintenance and support can be acquired for both FLOSS and proprietary. FLOSS can be supported through its project boards, but many businesses decide to pay for support. In this case support can be more or less expensive than for the proprietary software; the good news is that the support can be competed, tending to lower prices for support.

Cybersource found that the TCO tended be lower with FLOSS.

An InfoWorld survey of Chief Technical Officers (CTOs) found that they were saving money; 32% (about 1/3) say they were saving $250,000 or more a year.

InternetWeek/InformationWeek's survey of companies found that most believed that FLOSS cost less, anywhere from 25% to 75% less than proprietary programs.

TCO: Specific Examples

- **Measured Web server TCO of GNU/Linux is 40% (<1/2) of Windows' and 14% of Solaris' [RFG]**
- **Amazon.com: $17M savings in 1Q via Linux**
- **UK Gov't Becta* 3yr study: FLOSS savings significant in primary & secondary schools**
  - **Secondaries reduce IT overheads by 24% (inc. software, hardware, and support costs)**
  - **Primary schools cut computer costs by nearly half, primarily from support but also hardware**
- **Willamette U. Library $41K vs. $100-150K using networked X terminals [Murphy]**
- **Netproject: Desktop Linux 35% (save 65%!) of Windows**
- **Largo, FL: $1M/yr savings thin clients**

*Becta: British Educational Communications and Technology Association

18

Above is a list of some of the many who have reported saving money using FLOSS. Amazon.com credits this with saving their business; the savings were enough to turn losses into profits.

Largo, Florida is an interesting case; they report a $1million/year savings due to FLOSS, deployed using thin clients.

I no longer try to keep up with the many individual reports of money saved by particular organizations by using FLOSS. There are simply too many such reports to track.

Non-Quantitative

- **To many, non-quantitative advantages of FLOSS are more important, e.g.:**
  - **Social/ethical/moral reasons**
  - **Avoids risks of single source solutions/lock-in**
    - **Create reversible decision: can switch/self-support if price jacked up, maliciously changes interface, drops support, needs change (can get data), …**
  - **(Can) avoid security risks of monocultures**
  - **Supports domestic IT infrastructure**
  - **Long-term data retention (format not secret)**
  - **Many believe it encourages innovation**
  - **Avoids license management and litigation**
  - **Greater flexibility**
    - **Can change software (inc. via hiring) to meet needs**

19

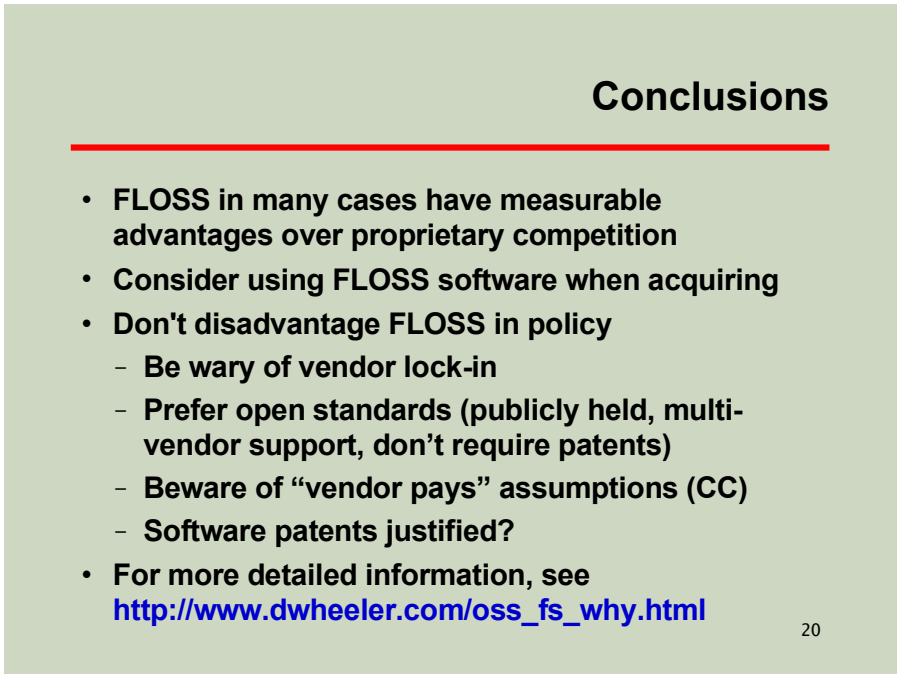
This is a survey of quantitative studies. However, before I close, I do want to clearly state that many people find some of the non-quantitative reasons for using FLOSS to be more important.

The Free Software Foundation and others have long argued that choosing Free Software is necessary for social, ethical, and moral reasons.

Clearly, FLOSS avoids risks of a single source solution or lock-in. If the support becomes too expensive, you can always switch, since you have the code necessary to perform that switch.

There have been a number of studies giving good evidence that FLOSS encourages innovation. I do not have time to dig into that in this presentation; see my paper for more such information.

Some other non-quantitative reasons are listed above.



Clearly FLOSS in many cases are measurably equivalent to, or superior to, their proprietary competition. Thus, when you are acquiring software, you should consider your FLOSS options. Since FLOSS rarely respond to requests for proposals (RFPs), you should certainly not assume that posting an RFP will get the best options; you may need to search for your best options.

If you are involved in setting policies, you should be careful to not disadvantage FLOSS in policy. Be wary of vendor lock-in, which may impair your ability to switch to a better alternative if one is available. Instead, prefer open standards (ones that are publicly held, supported by multiple vendors, and don't require patents to implement). Beware of "vendor pays" assumptions—those may not be reasonable for FLOSS projects. And indeed, all of this raises doubts about the value of software patents, which disproportionately harm FLOSS projects. There is no evidence that they actually help the software industry, but a great deal of evidence that software patents are fundamentally harmful.

For more information, see the paper at http://www.dwheeler.com/oss_fs_why.html

Backup Slides

- Introduction to FLOSS
  - Basics, history, OSS vs. FS, licenses, development model
- Unnecessary fears
- Acronyms
- Interesting sites/documents

Basics of FLOSS: Free-Libre / Open Source Software (OSS)

- **Free-Libre / Open Source Software (FLOSS) programs have licenses giving users the freedom:**
  - **to run the program for any purpose,**
  - **to study and modify the program, and**
  - **to freely redistribute copies of either the original or modified program (without royalties, etc.)**
- ***Not* non-commercial, *not* necessarily free-of-charge**
  - **Often supported via commercial companies**
- **Synonyms: Libre software, FLOS, OSS/FS**
- **Antonyms: proprietary software, closed software**

History of FLOSS

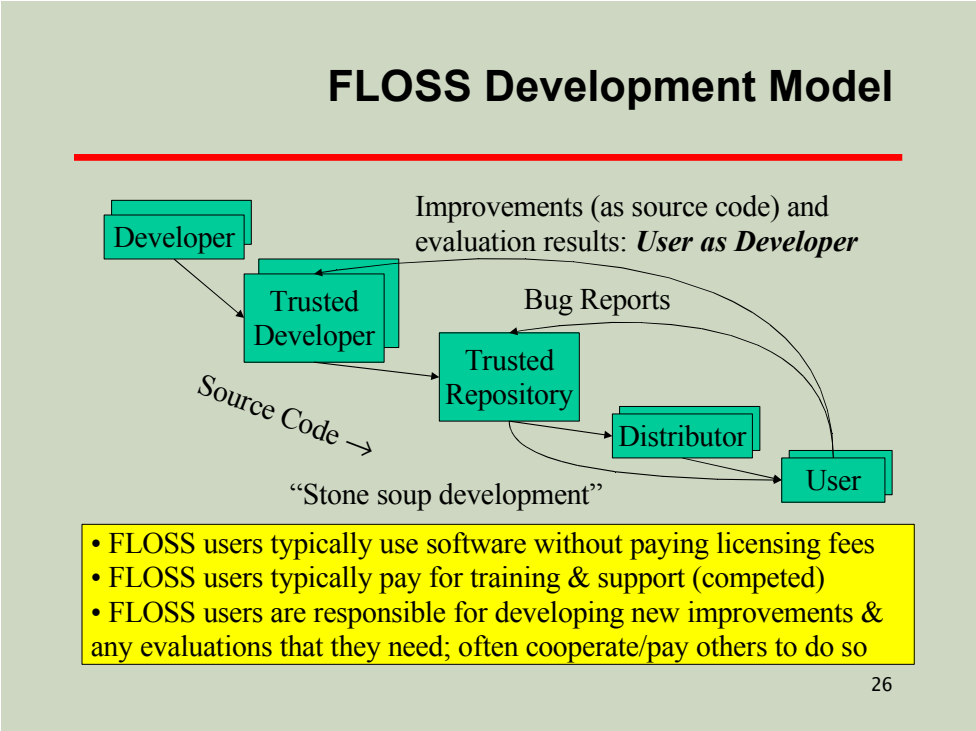
- 1950s, 1960s: Software freely distributed
- ~1970s: Rise of proprietary software
- 1984: Richard M. Stallman establishes "Free Software Foundation", creates "General Public License" (GPL)
- 1990s: Increasing Internet availability enables developer coordination
- 1997: Eric Raymond's "Cathedral & the Bazaar" explains new approaches; term "Open Source Software" coined

"Open Source Software" vs. "Free Software"

- First named "Free Software" by Stallman
  - Free as in Freedom
  - Officially defined by "Free Software Definition"
  - *Not* necessarily zero price; confused many
- New term coined: "Open Source Software"
  - Officially defined by the "Open Source Definition" (a long 9-point list)
  - Practically all OSS software is also FS
- Terms sometimes indicate motivations
  - FS: emphasize ethical/social issues
  - OSS: technical superiority/flexibility
  - OSS often used due to "zero price" confusion

Major FLOSS Licenses

- Many licenses, but 4 dominate
- BSD-new & MIT license: anything but sue
  - Can incorporate code into proprietary software
  - Financial incentive to use, but *not* aid project
- General Public License (GPL): "Copyleft"
  - If distribute, *must* distribute source code or provide written offer to do so
  - Cannot link (embed) into proprietary software
- Lesser/Library GPL - a compromise
  - Must distribute source code/written offer, but only of component itself
  - *Can* link into proprietary software
- Public domain is FLOSS, but rare



This figures shows a typical development model for OSS?FS. Arbitrary developers (who are typically users or represent users) submit their proposed changes to trusted developers for a given project. Those trusted developers are allowed to make changes to a trusted repository, the "master" copy used by others for the project. Users acquire release-ready versions of the master copy directly, or via 3rd-party "distributors" (who typically combine a set of projects and provide support).

Users can feed back suggestions and requests. A small subset of users may decide (for their own reasons) to make substantive changes or evaluations; such users begin to transform their role and become a new developer.

Typically FLOSS users don't pay for license fees, but they do pay for training and support. FLOSS users also pay (through time and/or money) for any new improvements, including any evaluations that they need.

Unnecessary Fears

- **Proprietary software always better supported? No.**
  - **Non-traditional support (mailing lists, etc.)**
  - **Pay for traditional support, and can compete it**
- **Proprietary more legal rights? No.**
  - **Who do you sue? Nobody, in either case**
- **FLOSS economically viable? Yes.**
  - **Many business models**
  - **Customers can band together**

Unnecessary Fears

- **Will programmers starve? No.**
  - **Estimated 95% software not developed for sale**
  - **Companies hire programmers to make changes for themselves**
  - **Widespread use of FLOSS moves software development into a service (not product) model**
- **FLOSS compatible with capitalism? Yes.**
  - **FLOSS development involves trade: code for code**
  - **FLOSS business often based on payment for support or commoditizing complements of products**
- **FLOSS mean no competition? No.**
  - **KDE vs. GNOME, emacs vs. vim**

Unnecessary Fears

- **Will FLOSS destroy intellectual property? No.**
  - **Usually, complaint is about GPL**
  - **GPL trades you the right to freely incorporate their code into your software in exchange for the right to freely incorporate your code [which incorporates their code] into theirs**
  - **Intellectual property traded for other intellectual property**
  - **Microsoft sells GPL'ed software, sponsored several FLOSS projects**

Unnecessary Fears

- Viewing and changing source code valuable for non-programmers? Surprisingly, yes.
  - "Would you buy a car with the hood welded shut? If not, what do you know about modern … engine technology?" [Bob Young]
  - Consumers demand this so they can have control over their product support, instead of dealers
- Anti-Microsoft campaign? No, not by all.
  - Jun02, 831 projects use Visual Basic; 8867 projects work on Windows [SourceForge]
  - Microsoft has been repeatedly asked to join community
  - Microsoft long used, and now develops FLOSS
  - Microsoft has sold GPL'ed software

Acronyms

- COTS: Commercial Off-the-Shelf (either proprietary or OSS)
- DoD: Department of Defense
- HP: Hewlitt-Packard Corporation
- JTA: Joint Technical Architecture (list of standards for the DoD); renamed to DISR
- OSDL: Open Source Development Labs
- OSS: Open Source Software
- RFP: Request for Proposal
- RH: Red Hat, Inc.
- U.S.: United States

Interesting Documents/Sites

- **"Why OSS/FS? Look at the Numbers!" (larger paper) http://www.dwheeler.com/oss_fs_why.html**
- **"Use of Free and Open Source Software in the US Dept. of Defense" (MITRE, sponsored by DISA)**
- **President's Information Technology Advisory Committee (PITAC) -- Panel on Open Source Software for High End Computing, October 2000**
- **"Open Source Software (OSS) in the DoD," DoD memo signed by John P. Stenbit (DoD CIO), May 28, 2003**
- **Center of Open Source and Government (EgovOS) http://www.egovos.org/**
- **OpenSector.org http://opensector.org**
- **Open Source and Industry Alliance http://www.osaia.org**
- **Open Source Initiative http://www.opensource.org**
- **Free Software Foundation http://www.fsf.org**
- **OSS/FS References http://www.dwheeler.com/oss_fs_refs.html**

Here are some interesting related documents and sites.

Presentation License

- **This presentation (C) Copyright 2005 David A. Wheeler. You may distribute unmodified and modified derivative copies; pick from:**
  - **Creative Commons Attribution-ShareAlike License version 2.5 or greater**
  - **GNU GPL version 2 or greater**
  - **GNU Free Documentation License (GFDL) version 1.2 or greater**
- **Please include a reference to http://www.dwheeler.com/numbers and http://www.dwheeler.com/oss_fs_why.html**