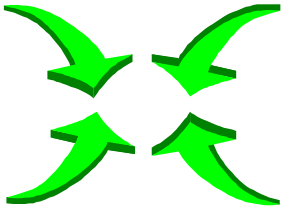
XVIII. Selecting a Computing Platform

Choosing Hardware, Software, and Networking Equipment
Criteria and Selection Process
Hardware Alternatives and their Features
Software Alternatives and their Features
Networking Alternatives

Computing Platforms

- A ***computing platform*** consists of the hardware, software (e.g., operating system, compilers, database management systems, etc.) and networking configuration.
- Key questions:
  - ✓What platform suits best the needs of the project?
  - ✓What criteria should be used to decide?
  - ✓How should hardware, software, and networking equipment be selected? Why?
  - ✓What general principles should be kept in mind?
- ***Key concern:*** Understanding of systems (hardware, software, and networking).

Major Platform Resource Types



- *Hardware*
- *Software*
- *Networking*
- *Human Resources*

Classes of Hardware Systems

- There are two basic classes of hardware systems: Commercial/ Business, and Technical/Engineering.

Large ↕ *Small*

Commercial/Business
- Mainframes
- Commercial Minicomputers
- Microcomputers
- Embedded systems

Technical/Engineering
- Supercomputers
- Workstations and Servers
- Microcomputers
- Embedded Systems

Commercial Minicomputers

- $10,000 to $500,000
- Proprietary hardware and software
  - ✓e.g. IBM AS/400, Compaq (DEC) VAX/VMS or Alpha/OpenVMS, HP MPE/iX
- Often "black-box" systems.
- Increasingly being replaced by UNIX- or Windows NT-based systems.
- Multi-user machines.

Workstations and Servers

- $5,000 to $500,000
- "Open" Systems, e.g. Sun SPARC, IBM RS6000, HP/UX, SGI/IRIX, Compaq Tru64, SCO, etc.
- Often at least partially "glass-box" systems.
- Encroaching on Minicomputer marketplace. Internet servers are key application area.
- Intrinsically multi-user-capable machines.

"Open" vs Proprietary

- The "Openness" of a system is a measure of the ease by which one system can be replaced by a another from an entirely different manufacturer.
- Proprietary systems available only from one vendor.
  - ✓Advantages: vendor provides everything!
  - ✓Disadvantages: price gouging, difficult migration, poor interoperability between vendors.
- "Open" systems are available from many vendors;
  - ✓Advantages: easy to change vendors, lower prices good interoperability.
  - ✓Disadvantages: minor differences, finger-pointing.
- Few systems are entirely proprietary or entirely open.

Black-Box vs Glass Box

- A "Black-Box" system is one in which only the vendor or integrator has access to the internals of the system; customer/user customization is limited. Often used for single "turn-key" applications.
  E.g. point-of-sale, bank ATM.
- A "Glass-Box" system has internals that are partially or completely accessible to the customer/user. Often preferred for general-purpose use.
  E.g. engineering workstation.
- Very few commercial systems are entirely "Glass-box".

Microcomputers

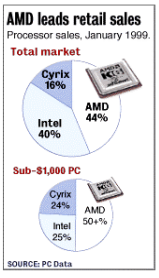
- $1,000 to $25,000.
- Generally "Open" hardware, proprietary software. Hardware becoming a commodity.
- Encroaching on Minicomputer and (to a lesser extent) the workstation and server marketplace.
- Wintel (WINdows on inTEL) are the main players in this category; Apple competes in niche markets.
- Generally single-user machines.

Wintel: MS Windows + Intel x86

- Microsoft Windows NT
  - ✓Server and desktop
  - ✓Window system on top of VMS-derived OS.
- Microsoft Windows 95/98/00
  - ✓desktop only
  - ✓Window system on top of DOS.
- Both proprietary, single-user.

Wintel: MS Windows + Intel x86

- Open architecture
  - ✓CPUs available from Intel, AMD, and Cyrix (despite Intel Slot 1 patent).
  - ✓Motherboard chipsets available from Intel, VIA, SiS; motherboards available from many vendors.
  - ✓Peripherals available from many vendors.
- Intel x86 hardware increasingly becoming a "commodity".



Apple Macintosh

- ~5% of the microcomputer marketplace.
- Niche markets: K-12 education, graphic design.
- Proprietary hardware and software.
- New inroads into home marketplace (iMac)
- Rarely chosen for office/business use.

UNIX Derivatives on x86

- Hybrid market: Workstation/Server-class software on x86 hardware.
- Both Commercial and Free options exist.
  - ✓Commercial: Sun Solaris x86, SCO Unixware, BSDI.
  - ✓Free: Linux, FreeBSD, OpenBSD, NetBSD
- Free UNIX derivatives (Linux, BSD) on Intel x86 with source code are true glass-box systems.
- Commercial vendors for Free UNIX exist.
- Inroads being made in both microcomputer and workstation marketplaces.

Java and Network Computers

- Java concept: compile-once run-everywhere.
- Network computer concept: stateless small microcomputer auto-downloads and uses software and data as needed from network.
- Java popular for network computers.
- Java used on wide range of systems, from workstations to embedded systems.

Networking

The Network is the Computer - Sun Microsystems

- Networking is a major component of almost every modern computer system.
  - ✓Client-Server
  - ✓Internet and Intranet
  - ✓Remote Office / Work-from-home
- Key Networking Areas
  - ✓Local Area Network = short-distance (one building)
  - ✓Backbone = medium-distance (campus)
  - ✓Wide Area Network = long-distance
  - ✓Remote Access = via phone/cable TV/satellite

Local Area Network (LAN)

- Connects machines within a single building/group
  - ✓Ethernet (copper)
    - ✓Available in 10/100 Mb versions (1Gb using fibre)
    - ✓Inexpensive, widely used.
  - ✓Token Ring
    - ✓Available in 4 and 16Mb versions
    - ✓Rarely used except by IBM
  - ✓ATM (copper)
    - ✓155 Mb (622Mb using fibre)
    - ✓Expensive, complex, flexible, high-overhead.
- Switch or shared hub.

Backbone Network

- Connect multiple groups, buildings.
  - ✓100Mb (fibre) or Gb Ethernet
    - ✓Fast, inexpensive, simple
  - ✓FDDI
    - ✓Old 100Mbit technology, increasingly obsolete
  - ✓ATM
    - ✓155Mb, 622Mb
    - ✓Expensive, powerful, flexible, complex, high-overhead
- Switched or Routed

Wide Area Network (WAN)

- Networking over long distances
  - – ATM, Frame Relay, T1/T3, etc.
- A WAN interconnects LANs/backbones in different locations, e.g. different cities.
- Internet is an interconnected set of public WANs
- Long-distance lines typically leased from e.g. telephone company.
- Satellite link sometimes used.
- Usually routed.

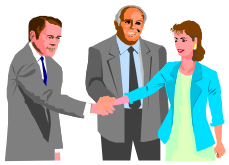
Remote Access

- Accessing a LAN or the Internet via phone/cable TV service.
  - Work from home
  - Access when travelling
  - Home internet service/personal web browsing
- Usually PPP over modem or cable modem.
- DSL services now available in some places.

Hardware and Networking

- Minicomputers -- Multi-user machines with terminals; networking often not needed.
- Workstations/Servers -- Extensive use of networking; LAN almost always an integral part of the system. Almost any application can run across the network.
- Microcomputers/PCs
  - ✓Stand-alone at home, (usually) networked at work.
  - ✓Single-user, hence require file servers for data sharing. Some (client-server) applications use network.

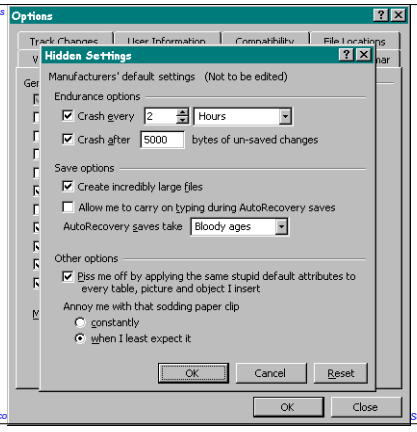
Human Resources



- Staff are ***most important consideration*** for any computer system for:
  - ✓Expertise availability
  - ✓On-going cost
  - ✓Training
- System admin, developers, documenters, operators, webmasters, help desk staff.
- Staff-related costs can often exceed HW, SW, or NW costs.

Systems and Human Resources

- Commercial Minicomputers
  - ✓Turn-key systems use vendor/integrator expertise.
  - ✓But customization may be expensive.
- Workstations and Servers
  - ✓Generally require UNIX-related expertise; when available, expertise often expensive.
  - ✓But people with deep OS knowledge easier to find.
- Microcomputers
  - ✓Basic expertise (filling in forms) widely available
  - ✓But deep OS knowledge often hard to find outside of MS, Apple.



Putting Everything Together!

How to make selection decisions?

What criteria to use?

How to avoid common pitfalls?

Understand Your Own Needs

- Hardware resources needed: CPU, memory size, memory bandwidth, I/O, disk space, etc.
- Software/OS resources needed: application availability, OS scalability
- Networking resources needed: network bandwidth, latency, remote access.
- Human Resources needed: OS expertise, hardware expertise, system administration requirements, user training/help desk requirements.
- Other needs: security, reliability, disaster recovery, uptime requirements.

Understand Your Own Resources

- Available funds
- Existing hardware, software, and networking.
- Existing staff and their expertise
  - ✓ Adding a system for which existing staff have no expertise requires training and/or hiring.
  - ✓ A new system that is difficult to administer may have a negative impact on existing staff.
- Special relationships or similar considerations: is there a special relationship with a vendor, reseller or other third party which makes a particular system choice more attractive?

Understand Available Options

- Are desired resource requirements feasible on a given system? Are they feasible only at great expense?
- What is the up-front cost of an appropriately configured system? What is the ongoing cost? What impact will it have on existing resources?
- What is the project schedule? How quickly can it be implemented? Will it be implemented in time?
- How mature is the proposed system technology? Is it too new to be stable? Is it likely to become obsolete soon?

Understanding Minicomputers

- Expensive to buy, expensive to maintain (vendor has customers "locked-in").
- If turn-key solution from vendor or reseller is available, very little staff expertise may be required.
- Computing model is generally a single multi-user machine or small cluster of such machines, with connected terminals. Network is often incidental, although increasingly used to connect terminals to machine(s).
- Central machine resources are more expensive, but one big machine is easier to administer than many smaller ones.

Understanding Workstations

- Relatively expensive but powerful. Multi-vendor availability allows mixing and matching equipment from different vendors, but with some staffing costs.
- Both minicomputer (single multi-user machine) and microcomputer (many single-user machines) models are feasible, as well as combinations of the two.
- Networking decision is critical, because of the key role of networking in this type of system.
- Expertise availability a major consideration.
- Servers can support PCs as well as workstations.

Understanding Microcomputers

- Hardware is very inexpensive and widely available.
- Resource requirements need to be carefully considered because of software and hardware limitations.
- Machines are single-user. "PC Servers" act as "network disks" or "network printers". Software runs on a single computer for a single user, unless specifically designed to be "client-server".
- Staff with basic knowledge of PCs is widely available, but deep knowledge is hard to find.
- PC administration can be difficult to automate, scale.

Understanding UNIX on x86

- If it runs a single-user operating system like Windows or MacOS, it is a PC -- not a workstation -- even if the vendor calls it one.
- If the same hardware runs a UNIX derivative or other true multi-user OS, it is a workstation even if the vendor calls it a PC.
- Windows NT capable of being a multi-user operating system, but is rarely configured or used that way.
- UNIX on x86 (e.g. Linux) shares the expansion limits and low costs of the PC world and the flexibility, networking issues and staff expertise issues of the workstation world.

Understanding Java and NCs

- Technology is new and relatively immature.
- NCs are cheap but require more server resources than PCs.
- Java software has the potential to run equally well on NCs, PCs, Workstations and even minicomputers. This provides great future flexibility.
- Java expertise is in great demand.
- A network of NCs is very easy to administer. All critical state is in one place (server).

Understanding Networking

- If remote access is an issue, note limited bandwidth.
- Use switched ethernet over copper for LAN, over fibre for backbone, unless there is reason not to.
- Choose WAN technology based on cost and availability.
- Minimize data transfer over remote access lines (modems) and WAN links as much as possible.
- Across-the-Internet connections are not secure. Use encryption if secure data (e.g. passwords) need to be transmitted.

Important General Principles

- Minimize distributed state.
- Put the network between the components that communicate the least, especially so for slow networks.
- Compute ongoing costs as well as up-front ones.
- Schedule is important: there is no substitute for calendar time.

Minimize Distributed State

- "State" is valuable non-replicated data. If "state" is distributed across many machines, it is difficult to make sure it isn't lost (reliability) or stolen (security).
- Why? System administration and support; Reliability and robustness; Security.
- How?
  - ✓Use fewer, larger servers where practical; Centralize data;
  - ✓Centralize system administration;
  - ✓Invest in network so that central resources are more readily accessible.

Position Network Wisely

- Why? Network bandwidth is much less than bandwidth inside machine. Performance will suffer if components communicate mostly over a network.
- How?
  - ✓When doing heavy I/O, especially database I/O, avoid network disks (SMB, NFS).
  - ✓Do not use remotely heavily graphical applications.
  - ✓Use a large multiprocessor instead of several small machines when inter-process bandwidth is important.
  - ✓Remote Access is not always a substitute for a LAN.

Compute Operating Costs

- Why?
  - ✓ Ongoing staff, maintenance costs can dwarf up-front costs.
  - ✓ Ongoing costs are often ignored or downplayed. This is unwise.
- How?
  - ✓ Use staff salary and benefits to compute dollar costs of staff time.
  - ✓ Use estimated interest rate to approximate the value of future money in today's dollars.

Consider the Schedule

- Why?
  - ✓ In IT, delays can have enormous costs.
  - ✓ One cannot make up for missing time by adding people.
- How?
  - ✓ Consider both time and money costs for staff training, hiring.
  - ✓ Schedule realistically;
  - ✓ Avoid excessively complex systems;
  - ✓ Be skeptical of vendor marketing promises;
  - ✓ Be careful if a technology is new and/or immature.

Example: University Teaching Lab

- Ilike University wants to install an engineering teaching lab. They have good staff already, with considerable expertise. There is increasing interest in Java. However, money is limited.
- Options include Java network computers, engineering workstations and/or PCs. PCs are too inflexible, engineering workstations are too expensive, and Java NCs are too new to be good choices.
- Ilike U. buys a multiprocessor UNIX server where most of the data is stored, a fast network, a few workstations for specialized purposes, and a bunch of PCs, some of which are running Linux. Both PCs and workstations are served from the UNIX server.

What's Good About This Solution?

This example shows the following:

- The effectiveness of taking into account one's existing resources (good staff).
- Combining multiple options (workstations, PCs) in clever ways.
- Taking advantage of UNIX on x86
- Centralizing "state" by putting it on central UNIX server.
- Positioning network wisely by providing a few workstations for graphically intensive workloads and a multiprocessor server for parallel scientific and/or I/O intensive workloads.

Example: Furniture Design

- The Siddown chair factory in Whatchamacallit, Yukon builds quality chairs for shipment all over the world. Unfortunately, the Avaseat chair company, their competitor, has just released a new line of chairs that could put Siddown out of business. The company decides to buy computers for their designers (6), to help them design a new line of chairs more quickly.
- The Siddown company decides to buy six Windows NT PCs with CAD software, a small ethernet switch and a Windows NT fileserver, where all the chair designs are stored. While workstations are often used for CAD and could potentially do the job well, UNIX expertise is hard to find in Whatchamacallit, and Siddown would have to pay high salaries to bring in someone from outside, while training someone locally would take too long.

What's Good About This Solution?

This example shows the following:

- Taking into account the schedule for the project (e.g. the enormous opportunity costs, in this case the bankruptcy of the business, if there are delays).
- Taking into account the availability of human resources (e.g. it can be hard to find UNIX experts in the Yukon).
- Centralizing state on an NT server rather than distributing the designs among the different PCs of the different designers.

Example: Warehouse Inventory

- The Keepawhile storage company has been very successful, and now needs a new inventory system for their four warehouses in two cities. All data needs to be accessible at all locations.
- Rather than keep a separate database in each warehouse, the Keepawhile company decides to buy a single machine, a turn-key inventory system running on an IBM AS/400. They use leased telephone lines and modems to connect terminals in all four warehouses to the central machine. The fact that the system is turn-key allows them to run it with a minimum of staff. The ongoing staff cost savings far outweighs the initial cost of the system or its ongoing maintenance cost.

What's Good About This Solution?

This example shows the following:

- Cost savings of taking human resources into account.
- Centralizing state/data.
- Positioning "network" wisely (in this case, avoiding a network altogether and putting everything on one machine).

Think for Yourself

- "Jumping on the Bandwagon" is endemic to IT.
  - ✓1979 industry quip: "Nobody ever got fired for buying IBM".
  - ✓1999 version: "Nobody ever got fired for buying Microsoft".
- Why? Ignorance, fear, hybris, misleading marketing, lack of management accountability.
- Pick what works, not what "everyone else is doing".
- Study all available options, not just the trendiest.
- Question vendor and "industry analyst" claims.