SYSTEM SELECTION.

The overall objective of system selection is to get the best system at the lowest cost.

1. SYSTEM REQUIREMENTS

Before you can decide on a system, you must decide what the system must do. This can be established by:

- a comparitive study with a similar organisation and applications
- hiring a consultant
- using a computer bureau { no hardware capital; offsets the cost of computing; some staff may be trained
- consideration of past/present
  find out (by asking questions and measuring present usage) what is presently being done in terms of programs (systems) processed and facilities required then predict future requirements from this. New applications can be considered by analysing similar applications, simulating the new application, or hiring a consultant.
- considering the type of environment - a fast changing environment needs high flexibility and expandability, while a slow changing environment allows easier prediction of future requirements.

Once the requirements have been determined, they should be giving priorities and split into mandatory and desirable requirements.

2. Mandatory vs. Desirable Requirements

2.1 System Cost

The greatest temptation is to impose a fixed upper bound. Rather use cost as an evaluation guide, because:

(1) often a system will be offered at your limit irrespective of whether you need all its facilities

(2) you can often get a much better system for a slight increase in price.

2.2 Throughput and Computational Speed.

Throughput is a better measure than speed (except possibly for computation bound jobs).

Speed is easy to measure and use to compare systems but is not the best method.

Comparisons of system performance:

- depends on the measure used
- a change in environment can change the performance of the system quite dramatically.

2.3 System Architecture (Seldom mandatory)

It is often immaterial if 0/1/2/3 address machine. Micro-programming could be an issue, but should be assessed under speed.

2.4 OPERATING SYSTEM (Seldom mandatory)

Beware of limitations, particularly of program size, with regard to paging, segmentation, virtual memory, etc.

2.5 DELIVERY DATES (Seldom mandatory)

The use of the existing system can usually be extended if necessary, and even if funds have to be committed by some date before delivery, a negotiated agreement is usually possible.

2.6 COMPATABILITY.

The conversion of existing software is very important. Don't overestimate the old progs and out of date system, or underestimate rewrite cost vs. conversion cost/time. Possibly micro-coded simulators can be used during the intermediate period.

2.7 APPLICATION PACKAGES.

Be wary of insisting on a specific package. High level language packages are often very portable, but not data base packages.

2.8 MANUFACTURER SUPPORT

Consider training, backup systems, compilers, access to a system before installation (desirable from manufacturer, mandatory in any case)

2.9 RELIABILITY

This is virtually impossible to specify as mandatory, but can possibly be tied to some predefined penalty clause.
Reliability can be specified by:
- % uptime
- no of breakdowns
- in terms of the entire system or individual units.

2.10 SPACE REQUIREMENTS. (Seldom mandatory)

Sufficient space should always be available.

2.11 I/O REQUIREMENTS

Ensure that these requirements are realistic. Often it is response that should be mandatory. Also consider upgrading & expandability.

3 DETAILS OF SPECIFICATION TO VENDOR

A general specification means:
- vendor can make optimum use of best features of his system
- often a cheaper system emerges
- the evaluation is more complex.

A detailed specification means:
- little leeway for vendor
- simplifies design work for vendor
- simplifies selection of system
- system is no better than spec.

4 ITEMS TO BE CONSIDERED

4.1 COSTS (see handout pg 3)

For each proposed system calculate total cost:
- one time costs should be spread over planned system life
- include cost of any planned expansion
- itemise individual equipment costs
- take care not to duplicate costs.

4.2 SYSTEM CHARACTERISTICS (see handout pg 4)

Evaluate these in terms of system throughput and increased programmer / analyst productivity

4.3 EXPANDABILITY (see handout pg 5).

This is much more important in a fast-changing environment.

4.4 VENDOR SUPPORT

5 TECHNIQUES FOR EVALUATING SYSTEM PROPOSALS

5.1 SINGLE MANUFACTURER/

Often used because:
- satisfaction with equipment and support given in the past
- minimisation of existing program conversion cost
- minimisation of staff retraining
- little effort in evaluating proposal.

Disadvantage: may not be the best, cheapest system.

5.2 Overall Impression.

Subjective impression gained by:
- reading proposals
- visits to existing installations - ask if they are satisfied

This requires little detailed analysis and is cheap and easy, but is not very objective and may result in buying from the best salesman instead of getting the best system

5.3 Cost.

Ensure system meets the mandatory requirements then choose the cheapest system. This is easy to choose, but a slightly more expensive system may satisfy more desirable requirements and it is easy to underestimate expansion capabilities of a system, as it is often not a d cost item.

5.4 Weighted Scoring.

From the systems satisfying the mandatory requirements, select the system which rates highest on weighted scores for the desirable requirements. The difficulties with this are establishing meaningful relationships between the number of points for high performance and those for low cost. The weights should be preassigned to eliminate subjectivity

5.5 Cost - Value Technique.

- Validate mandatory requirements.
- For each desirable feature assign a money value and sum these ΣV
- Total system cost value (TSCV) = total system cost $- \Sigma V$
- Choose system with lowest TSCV.

5.6 Evaluation Guides

We need to get a predetermined value for each desirable requirement independent of vendors' proposal.

5.6.1 General Concepts Used.

Upper Cost - Value Limit

How do we estimate value? As a first approximation, we can equate value with cost. This might be highly misleading, however. (eg a math S/R may cost R10 000, but be used by only 2 people ⇒ value more like R500)

Diminishing Value of Large Amounts.

Concept: as more of an item becomes available its value decreases (as if really pushed you could do with less, so the extra capacity has less value) The reduction curve chosen is somewhat arbitrary

Example

- Cost of system R1-million over 5 years.
- Original spec 20% load increase / year.
- mandatory system life time 5 years.
- 4 systems proposed with different expansion potentials beyond that necessary
  - (a) we depreciate each extra year of system life as :- a change in environment could be faster than expected
    - new technology may become available at a cheaper cost.
  - (b) add in cost of additional equipment to give any additional expansion

(See graph A on handout).

  - (c) different depreciation functions can be chosen. They should have the same gross behaviour.

5 Methods for Aquiring Computer Systems

5.1 Introduction

- Best method depends largely on the financial set up
- Price
- Tax write offs?
- Availability of money
- Financial analysts must be involved in decision
- Also depends on:
  - (a) vendor proposals
  - (b) accuracy of your prediction of the computer system environment (when you can't predict long use rather lease than buy).

The three most common aquisition methods are:
- leasing
- purchasing
- leasing with purchase option.

Note that the manufacturers price policy can change at short notice, and that the price can change from customer to customer

5.2 LEASING

- User pays a predetermined monthly price for some pre-determined no of hours of usage.

- Price (rent & maintenance)
- Rent (cost of finance for equipment, and profit)
- Maintenance (cost of parts & labour)

- Leasing contract conditions:
    - can affect when you upgrade (minimum period of usage for each item may be specified)
    - cost of extra usage (above pre-determined amount) can be high.

5.3 PURCHASING

Advantages:
- use of system whenever you want
- maintenance can be done by:
    - vendor
    - independent service organisation (pref established)
    - own maintenance staff.

- Vendor has to state whether equipment is new, second-hand or refurbished (can be better than 2nd hand)

5.4 LEASING WITH PURCHASE OPTION.

- The option is usually taken up within 1-2 years
- A stated % of rental paid to date (usually small) is deducted from full purchase price.

Ads
- user has a trial period with equipment running it in actual environment
- vendor has to state if equipment is new or refurbished.

5.5 LEASE - BACK PLANS.

- An intermediate company buys equipment and leases it to you.

Ads
(a) cheaper
(b) maintenance by leasing company possible
(c) sub - leasing time is possible

However, this usually means a longer lease.

5.6 SPLIT AQUISITION.

By negotiation
- user owns part of system (say CPU) and rents the rest of the system

Ad: can lead to using cheaper plug compatible peripherals

5.7 Second-Hand Equipment.

Cheaper, but more prone to obsolescence.
Check: - maintenance costs
- length of maintenance contract
- what software support is available
- reliability history of the particular equipment.

6 Validation of System Capabilities and Performance.

You have to check the information given to you about a system.
- the good points of any proposed system will be emphasized to you by sales personnel.
- user should validate
  (a) system characteristics
  (b) system performance.
- the vendor must meet his claims at the cost specified (? in the contract)

6.1 General Validation

System capabilities can be verified by reading and understanding the technical documentation, but this is - time consuming
- possibly difficult to understand
- not all required info may be there.

- Study Auerbach Computing reports

A better approach is to use a questionaire. It obtains the desired information in a uniform way, and compels vendors to give information that may otherwise be omitted.

6.2 CONTENTS OF A COMMON QUESTIONAIRRE

- speed
- compatibility
- expandibility
- environmental requirements
- capacity
- reliability
- software

6.3 SYSTEM PERFORMANCE EVALUATION (SPE)

Evaluating performance is difficult
Techniques: - instruction timing
- application benchmarks
- computer simulation

- main difficulties are not so much with the method but with the conclusions to be drawn from various performance measures

The goal of SPE is the prediction of overall system performance. There are no generally accepted criteria, but throughput and terminal response time can be checked for some systems.

6.3.1 INSTRUCTION TIMINGS

GIBSON MISC.

Concept: - analyse a large number of programs already being used

- determine occurence of the various machine instructions
- use the time of the new machine to calculate the new time required to process program.

Difficulties: No account taken of:
- memory size
- word length
- I/O estimation - o/s interference with I/O.

6.3.2 POWER METHOD.

(a) GROSCH'S LAW

$$\text{POWER} = \text{CONST} * (\text{SYSTEM COST})^2$$

This is a rough guide that was useful, but has limits to economy of scale. Small cheap machines have undermined the law.

(b) RAUBTEN AND MEEK

$$\text{POWER} \propto \frac{\text{MEMORY SIZE} * \text{WORD SIZE}}{t_{cpu} + t_{I/O}}$$

t_{cpu} = time for cpu to do 1-million calculations.
$t_{I/O}$ = time cpu is idle waiting for I/O while performing 10^6 calculations.

No measure of software (o/s, compilers) is included
Ads - Can be done for very little cost
- Allows systems to be compared independant of software.

6.3.3 BENCHMARKING.

These are standardised programs used to evaluate and compare the processing capability of different systems. They specify:

- input
- computations (the specific method used can be left up to the vendor so he can use his system to the full)
- output

6.3.4 STANDARD AUERBACH EDP REPORTS.

Consists of 5 benchmark tests.

① Generalised file processing
② Random access file processing
③ Sorting
④ Matrix inversion
⑤ Generalised math problem.

This has allowed a general comparison between machines since 1962.

Mistake see sec 8 for continuation of this section.

7 Contractual Issues

Several suppliers may provide the required facilities at an acceptable cost. It can be as important to get the right deal as it is to get the right system.

7.1 Misconceptions

1. Any contract offered by a 'reputable' supplier will be fair and reasonable
   - most standard contracts do not protect the customer in important ways.

2. A contract is just a piece of paper signed when the order is placed
   - the contract includes all binding commitments, rights and obligations of the parties

3. Contract can be left to customer's legal department
   - computer suppliers specialise in agreements, customers normally don't.
   - executives who use computer & legal dept must liase re. contract.

4. First choose a supplier, then negotiate details of contract.
   - once a supplier has been chosen, customers negotiating power vanishes
   - major contract details should form part of the assessment process.

(5) Terms of contract are irrelevant as any later need of them is evidence of a major failure.
- most standard contracts do not safeguard buyer
- friction occurs when parties have a different interpretation of what was meant.
- rather define them first, then less misunderstanding will occur.

(6) small print - read it.

7.2 SOME BASIC CONCEPTS.

- object is to help customer obtain proper benefits from his legal advisers and to aid him in negotiations with suppliers.
- not a substitute for seeking proper legal advice
- not a treatise on law of contract.

7.2.1 FITNESS FOR PURPOSE

- Obviously bears on capacity of equipment to do required work
- Purpose should be clear (possibly tied to specs of system)

7.2.2. MERCHANTABLE QUALITY (ie reliability)

7.2.3 CONSEQUENTIAL LOSSES.

- suppliers must rectify fault and compensate customer for loss / damage suffered as a consequence
- difficult to prove and quantify, so often liquidate damages are preferred (ie predetermined figure

agreed to if certain contingencies occur - without proof.

7.2.4 EXCLUSION CLAUSES.

- Majority (but not all) suppliers, as standard practice, exclude these rights in their terms & conditions
- Often consumer protection legistlation invalidates these, but generally protection is not extended to capital goods.
- Often modest warrantee given (ie in working order for some period after delivery) and will then go on to expressly exclude all other warranties.
- Attempt to get a contract that does not contain blanket clauses (but possibly containing limitation of liability amounts)

7.2.5 SOLE AGREEMENT

- Often contract will contain the seemingly innocuous statement that it is sole agreement.
- Very dangerous if customer's statement of requirements is thereby excluded.
- Also any other written obligations.

7.3 JUSTIFICATION FOR EXCLUSIONS

- Essentially none
- Some excuses given (clause means something else??, clause does not matter, anti-trust measures require them)

7.4 Acceptance Criteria

- Specify these in contract.
  - operates according to manufacturers' spec.
  - for both hardware & software
(check airconditioning, elec, etc)

7.5 Some ways of checking that specs were met.

- In well defined environment run benchmark progs for some time (say 72 hrs) to asses performance.
- Run normal workload with requirement all subsystems up for (say) 95% of normal working hours for (say) 20 working days.

7.6 Some Problems.

(a) Definition of system down time

Is it total system down, or an individual unit? Whichever is decided on do it before and incorporate into contract.

(b) What if system does not meet acceptance test?

- decide this before hand
- give vendor some time to remedy situation especially if some subsystem only is malfunctioning
- software acceptance criteria difficult.

8. More on Evaluating Systems.

8.1 Comments on Benchmarking.

- Routines must actually be run on the system
- Total throughput time is important
- Each routine is aimed at a specific application
- Results only meaningful if benchmark is well prepared
- Benchmarking can give info on effects of I/O, compilers, etc, on specific applications
- Benchmarks are affected by proficiency of programmer
- Different timings can be obtained from seemingly identical runs (eg due to different file placement)
- In practice, use benchmark progs that are representative of your workload.
- Split workload into classes (eg math, commercial, engineering, etc)
- Within each class choose 1 problem to benchmark and weight accordingly
- Use results from this benchmark to predict what will occur for the other problems in that class
- Each class will represent some % of total workload.

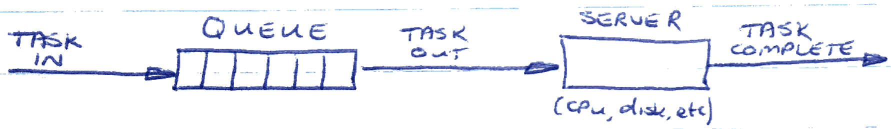
8.2 Synthetic Benchmarks.

- Parameter driven program
- Contains a module for each type of resource in the system.
- By suitable setting of the parameters any mix of CPU and I/O instructions can be chosen.
- To use this meaningfully you have to know (by analysis) what ~~this~~ type of mix to set up.

8.3 COMPUTER SIMULATION.

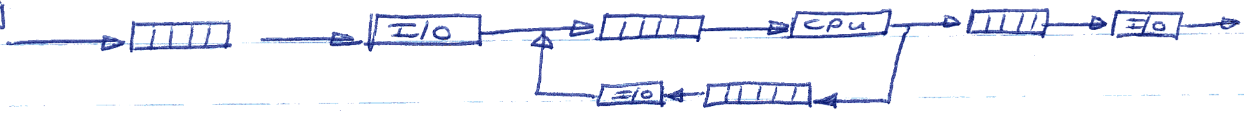
- You model the proposed system and then compare the performance of the various systems & configurations

- the basic concept used is:



- each active component of a computing system can be modeled in this way.

eg



NATURE OF A SYSTEMS MODEL.

- It is an abstract representation of the computer and is supplied with the workload of jobs.

Components:

1. Workload description (ie arrival & execution time of jobs)
2. System structure (components of system and paths between components that job can take)
3. Scheduling (rules specifying how jobs are selected for movement within system)

PERFORMANCE INDICES

(a) JOB STREAM PERFORMANCE

Indices include mean elapse (response time), throughput, frequency distribution of response times, etc. Those indices are orientated towards users who supply the jobs rather than to the system manager.

(b) RESOURCE PERFORMANCE

Indices reflect resource utilisations

GENERAL

- A simulation is highly technical - only use it if you have the expertise.

- Use a consultant who is a specialist in this area.
  - specialist packages are available
  - consultant will characterise system for you

Irrespective of whether a simulation model is used in the choice of machine it should be used throughout the life of the system to predict and measure performance.