Computation Research Group

CGTM-164
January 1975

COSTOPTIMAL RELIABILITY OF DATA PROCESSING SYSTEMS*

Kornel Terplan†

Visitor, Computation Research Group
Stanford Linear Accelerator Center
Stanford, California

(Submitted to ACM Performance Evaluation Review)

*Work supported in part by the Atomic Energy Commission under contract AT(043)515

†Permanent Address: Computer Center of the Association for the Hungarian Telecommunication Industry, Budapest, Hungary

1. Introduction

With the advent of third generation computing systems, the increase in complexity and power has reached a degree which exceeds the human ability to understand, to analyze, to predict, and to optimize system performance and reliability. The only method that can help is measurement. In defining measurement purposes, one has to define which measurable quantities in the system are significant and which may be ignored. But, at the present time, we do not know in general what is relevant in the measurements. For the sake of clarity, it is useful to define several levels of measurement.

- organizational level
- computer center level
- computing systems level
- job level
- computing subsystem level

At the computing system level, for example, we have to define the overall measures regarding reliability and performance. The intrinsic power (potential performance) of the system can be defined as the capacity to meet the requirements of a given workload. The efficiency of the computing system can be finally defined as ration between the actual and potential performance under the given workload. The measures are generally peculiar to the computing systems. However, some of them are of almost universal importance -- throughput and turnaround.* The first one takes into account the requirement of the "server"; the second one, more the demands of the user. Since most of the systems are subjected to widely varying loading conditions, it is not sensible to discuss throughput characteristics without simultaneous consideration of

*(10) uses similar approach; however, instead of turnaround or response, the general term "delay" will be used.

the turnaround (response) requirements. Thus, one can find that if the response time as a performance measure is allowed to increase, the throughput can be similarly increased (possibility of using powerful scheduling algorithms). The stable state can be achieved in the operating environment during system tuning, which means the process of measuring the system, understanding effects, and making small changes in hardware and software that cause large increases in system performance.[9] At this level of the approach, this is equally valid for batch processing and real time systems. In order to use quantitative methods in further chapters, it seems to be advantageous to use the following very much general measures for the reliability control of computing systems:

- throughput characteristics at given turnaround or response time
- quality of system outputs
- operating costs.

The "technical" reliability can be defined as the probability that the system will operate satisfactorily for a given period of time, under definite conditions. Pinkerton defines the system performance as the "probability of completing a given throughput of work in a given period of time.[11]

It seems to be very convenient to modify the reliability definition in order to meet the performance characteristics, too. An additional benefit consists of eliminating cases, where miscellaneous errors are ascribed after recording to some extent to real hardware or system software faults.

2. Computing System Reliability

In general, the productivity and reliability of computing systems depend on how well its hardware, software and human resources are selected and employed to do the work they are best suited for. Thus, as a first conclusion, we have to look at the system as a whole. From the view of the next higher

measurement and evaluation level (computer center organization), all system components (subsystems) have the same responsibility for overall reliability and efficiency loss. As a second conclusion we have to find the fundamental relationships for optimizing subsystems with regard to the rest of the system. With no claim to completeness and generality, Figure 1 can be used for the classification of the system elements.

To process data efficiently, with high reliability, the hardware components (CPU, main storage space, I/O devices, I/O channels, direct access storage space, speed of access, communication lines, etc.) of a computing system must be available when they are needed. The efficiency of the definite hardware components depends on the quality of system software (operating system). The program or set of programs that directs a computing system to perform definite operations is called an operating system, whose main parts are, in general:

- organizing or control programs
- processing programs
- utilities
- program-trials parts.

The hardware and software components are concerned with solving user problems. For successful solutions, the problem philosophy, the design method chosen, the quality and reliability of programs, and accuracy of data have to be taken into consideration. Furthermore, external and internal organization and human factors influence the system reliability.

Hardware errors are relatively rare today. Error detecting circuits and codes are frequently used for preventing these errors in the computer. Generally speaking, the hardware technology increases reliability, but the reliability is less than 100%. The errors can be grouped as follows: human-

machine interface, machine readable documents, terminals, data transmission, mass storage, and processing logic. Systems software, especially in its initial versions, sometimes has undetected errors. New methods are increasingly introduced for better controlling the reliability of software. The method of program testing is a crucial part of system performance.

The problem philosophy determines the effective use of computers in data processing systems from the view of a higher organizational level. The system can operate efficiently, disregarding effectivity. Inclusion of the "user" components (Figure 1) into computing systems performance makes effective systems operation achievable. When the computer seems to be wrong, the cause is normally an application program error. The solution to this problem is very thorough program testing, e.g., using automatic testing methods. The extent of testing depends on accuracy requirements and on cost considerations. Some of the sources are[7] inaccurate and/or insufficient user documentation and communication between specialists, lack of planning and of training, complexity of programming language, program errors (file description, inconsistent program specification, rounding errors, etc.).

The largest source of computer errors is not the failure of hardware and systems software, but data input. There are a large number of accuracy control strategies introduced and subdivided regarding the kinds of operations (batch, real-time, telecommunication networks). At the present time, it is very difficult to estimate the unreliability or reliability of human factors. Some of the sources are: error by creation, by omission, by reversal data, misunderstanding of instructions, misinterpretation of written material, deliberate errors, etc.

Using Figure 1, we have to calculate the sensitivity of components regarding the computing system reliability. All components of Figure 1 can be

characterized by the behavior corresponding to the parameters of the reliability which are throughput (T), quality (Q), and cost (C). For the summarized form, we get:

$$\begin{array}{ll} \text{Hardware} & \underline{H} = (T_h, Q_h, C_h) \\ \text{Software} & \underline{S} = (T_s, Q_s, C_s) \\ \text{User} & \underline{U} = (T_u, Q_u, C_u) \\ \text{Organization} & \underline{O} = (T_o, Q_o, C_o) \end{array} \qquad (1)$$

In the simplest approach, the components affect the overall performance and reliability in a sequential manner (in practical cases, it is more sophisticated while, for example, program design and program trial systems are not independent from each other, etc.). Equation (1) can be eliminated corresponding to the parameters (16 subsystems corresponding to Figure 1).

$$\begin{aligned} \underline{T} &= (T_1, \ldots . T_{16}) \\ \underline{Q} &= (A_1, \ldots . Q_{16}) \\ \underline{C} &= (C_1, \ldots . C_{16}) \end{aligned} \qquad (2)$$

In investigating the throughput and cost characteristics, T_i and C_k (i.e.(1,2...16, ke(1,2...16)) are expected to be relative measures given as percentage data.

Via equation (2), we are able to find the "bottleneck" equation characterizing the "worst" subsystems:

$$Z = (T_i, Q_j, C_k) \qquad (3)$$
$$i\epsilon(1,2\ldots 16),\ j\epsilon(1,2\ldots 16),\ k\epsilon(1,2\ldots 16).$$

Equation (3) guides us in how to improve system reliability in the most effective way. One of the tasks of compumetrics[8] consists of determining appropriate statistical techniques - including validation - to find the components of equation (2).

The prescribed parameters of the reliability have to be tuned during the system installation or improvement procedure. Being the parameter of dynamic character, it is possible to investigate trade-offs between the parameters. Some examples are:

- more sophisticated error checking techniques for quality improvement decrease running efficiency
- decreasing throughput causes more cost, e.g., in the form of upgrading required (computer center)
- decreasing turnaround time (response time) causes additional costs in charging for increased priority (user), etc.

Some trade-offs are investigated in (13), especially between throughput efficiency and costs of different kinds of memory.

3. Preventive Techniques

Prevention can be defined as a set of technical and organizational techniques which, when implemented, will reduce the occurrence or the probability of occurrence of disturbances. Some of these techniques are computing system elements with higher fundamental reliability, accuracy control, program testing, hardware and software maintenance, renewal, different kinds of redundance, structural and modular system and program architecture, etc.

The introduction of preventive techniques is very useful, for the following reasons:

- eliminating the need for other techniques (detection, error correction, etc.)
- earliest possible stage of error fighting, thus delay of unreliability
- costs are typically lower than detection and correction for achieving a given level of reliability (see later the "balance" equation).

The detection is a set of techniques which indicate deviations from the intentions or objectives of the system, and which reveal and spot the disturbances, using several methods (inspection, technical diagnostics, continuous monitoring, etc.). The correction techniques have to be planned as early as possible in order to recover, as fast as possible, from system disturbances. The components are speed, reliability, resource efficiency, data base, recovery strategy, etc.).

In order to control disturbances, the consequences (e.g., inability to process, loss of an entire file or single records of it, modification of records, unauthorized reading and copying, etc.) must be made quantifiable. For the purpose of quantification, the best way is to try to transfer all kinds of consequences into costs. Generally speaking, costs can be grouped according to:

- time of occurrence
- proportionality
- subdivision into time-dependent and time-independent factors

In accordance with the time behavior, we get the following costs:

- costs of stop run
- costs for standstill
- run up costs

All sources of errors and disturbances endanger the reliable way of systems operation regarding the parameters mentioned previously.

One can find the fundamentals of several optimization methods, dealing with the improvement of computing system's reliability and optimization, in the quite general equation which is to be minimized.

$$E[C_{total}] = C^{o}prev + f_A E[Y(prev^{o})] + f_D[E(Y(prev^{o})] \cdot E[X(prev^{o})] \quad (4)$$

Where the notation is defined as:

$E[\ ]$	Expected value
$(prev^{o})$	In function of using optimal preventive defense strategy
C	Costs
f_A, f_D	Constants depending on the category of costs $(f_A = \sum_{i=1}^{n} f_{Ai}; f_D = \sum_{j=1}^{m} f_{Dj})$
Y,X	Random (independent) variables, characterizing the duration and frequency of disturbances
i,j	Index
n,m	Number of different costs being considered

In order to evaluate equation (4), there are a large number of measurements required. In equation (4), all kinds of costs corresponding to detection, correction, and recovery are considered as consequence costs. It is assumed, evaluating the above equation, that the cost-optimal preventive defense strategy has already been found for several restrictions. The solution can be achieved in solving the so-called allocation problem of a given set of financial funds (restriction) by maximizing the system reliability (objective function). These kinds of problems can be solved by using dynamic or integer programming and simulation techniques. Equation (4) guides us as to how to achieve the cost-optimal system reliability when the bottlenecks (equation (3)) of the system have been discovered.

4. Advantages of the Reliability Concept

The approach introduced previously does not depend on the level of observation and evaluation; thus, it is useful for both approaches: macroevaluation (computing system as an entity) and microevaluation (investigation of the scheduling mechanisms of the CPU regarding throughput in a multiprogramming environment).

In the system improvement procedure (tuning), one is able to focus the bottlenecks of the system, and using equation (3), bottleneck shifting will not occur. The approach itself is not limited to the investigation of the fundamental measures only. Even in most practical cases, measures of greater detail are of importance (resource utilization factor, peak message capability per time unit, internal and external delay factors, transaction-throughput per time unit, etc.). In the macro approach, these are "compressed" into the throughput time.

For the first evaluations, some of the components can be put together (e.g., channels and I/O devices). The approach can be used for auditing the computer aided data processing system or the efficiency of the computing system (in the latter case, many users are considered corresponding to Figure 1; we can take either the average for rough evaluation or we have to consider more subsystems in equation (1)). Thanks to the modular structure, the concept is not limited to any special hardware/software combinations. Thus, using it for predicting system reliability and performance, new components (e.g., new mass storage technique) can be investigated.

It is very important to detect the relationships between variables. Sometimes it is difficult to determine the precise functions between general measures of the reliability and the variables measured in computing systems. Detecting correlations between two or more variables, one is interested in determining cause-effect relationships (equation 1: subsystem-behavior as cause and the measures as effect). However, correlations can be caused indirectly and analysis of the correlations source must be performed very carefully before exactly defining the interrelationships. Let us consider the following example:

Let us assume that after evaluating and eliminating equation (2) we get the following vector for characterizing the computing system behavior:

$$\underline{z} = (T_{14}, Q_1, C_6) \tag{5}$$

that means that in the assumed batch environment, the "worst" components are the following ones:

Regarding throughput: external organization

Regarding quality: data

Regarding costs: problem philosophy.

Based upon these experiences, we use equation (4) for improving the system reliability. In the over-simplified case study, the quality requirements are only considered. The main goal consists of determining the accuracy control strategy which minimizes equation (4). The incorrectness of input data will be corrected by appropriate modification of the source program. One has to pay attention so that the modified solution does not become critical regarding the other parameters of the reliability.

Fundamentally, input checks, in-process, and output checks can be used. Some of them are listed below.[3]

Input checks: Character (numeric, alphabetic, sign, etc.); field (limit, range, reasonableness, sequence, etc.); record (completeness, interval and external consistency, sequence, etc.); batch (control totals, brush totals, etc.).

In process checks: Arithmetic, rounding, reasonableness, artificial processing.

Output checks: Reasonableness, serial number tests, generating control records.

Introducing error detection and correction steps into the source program will cause several costs to be incurred, e.g., programming, debugging, increased memory size required, re-entry of corrected records, manual error

correction, increased throughput time, more statement, more error, etc.

On the other hand, costs can be caused by rerunning the whole job, increased maintenance costs, manual error detection, loss of users' confidence, etc.

Let us consider, for the purpose of optimization, a three-step model (input, in-process, output checks). The following notation will be used:

p_i reliability of step i for detecting all errors

p_i^1 improved reliability of step i for detecting all errors

$$p_i^1 = 1 - (1 - p_i)^{1 + u_i} \tag{6}$$

where u_i = f(preventive technique).

We have to maximize the following objective function:

$$R_3 = \prod_{i=1}^{3} p_i^1(p_1, u_i) \tag{7}$$

taking into consideration the following restriction:

$$\sum_{i\in 1}^{3} k_1(u_i) \leq C_{prev} \quad . \tag{8}$$

For clarity's sake, some comments are made about the procedure previously introduced:

- $k_i(u_i)$ are the costs for preventive defense
- equations (6) - (8) are concerned with solving the s.c. allocation problem in a parametric way for a set of C_{prev} values
- C_{prev} (eq 8) has to be taken into eq (4) as a first term
- all kinds of cost consequences have to be taken into eq (4) as a second term
- there are many measurements required for quantifying eq (6)
- generally speaking, detecting errors is easier at a later step, but correcting them is more expensive.

In preparing for future applications, we have to investigate several hypotheses concerning possible bottlenecks and their solutions. Some of them are listed below:

- modifying paging algorithms
- memory relocation in multiprocessing systems
- performance improvement by using multiple channel controllers[6]
- tuning the factor of interleaving
- investigation of problem program efficiency
- evaluation of data management strategies
- measurement and evaluation of mass storage devices, etc.

5. Data Base - Measurement Methods

In order to evaluate the equations introduced previously, one needs a large number of data. For the technical reliability aspect (DATAMETRICS[7]), one can use for evaluation of hardware and software the technical terms, such as rendundancy, probability for error occurrence, detection and correction, service and repairability, maintainability, security or attack repulsion, etc. The recording and analysis of errors and their detection and correction is very useful in evaluating and in calculating equation (4). Roughly speaking, these terms affect primarily the quality aspects, and secondarily the throughput characteristics of equation (1).

For the comprehensive reliability evaluation, one needs more sophisticated measurement and modeling techniques. The monitors are frequently used for these purposes. One of the possible categorizations can be found in Reference 9; Table 1.

TABLE 1

Basic Tool Categories

Category	Characteristics
Implementation medium	Hardware Software
Separability	Integrated Separated
Sample Portion	Full-time Sampling
Analysis concurrency	Concurrent Recorded data
Data presentation	Static statistic Time related

In general, simulation is a powerful tool applicable for analyzing existing systems and for predicting the reliability of new systems. The results of simulation, however, are very sensitive to the assumptions being made concerning the design procedure of the original model and data to test the model. Small deviations in either can cause large discrepancies between performance of the model and that of the reality. Therefore, when using simulation techniques, it is necessary to introduce validation techniques before utilizing simulation results. We can expect the wide use of trace driven simulation in the future.

6. Future Trends

For evaluating and solving the costoptimal reliability problems, there are a large number of measured data required. These data can be extracted by manually reporting system component errors, by frequently using monitoring techniques, and by measuring all system components.

In the performance and reliability evaluation field, several problems and questions have to be solved in the future. Some of them are:

- introducing standard measures of reliability and performance
- measuring the intrinsic power of computing systems
- unification of terminology
- interaction between data security and measuring
- interaction between compatability and bottleneck shifting
- integration of human characteristics in reliability, etc.

References

1. Kolence, K.W., Systems Measurement: Theory and Practice, INFOTECH State of the Art Report 1:1, p 391.

2. Drummond, M.E., Evaluation and Measurement Techniques for Digital Computer Systems, Prentice-Hall, Inc., 1973.

3. Martin, J., Security, Accuracy and Privacy in Computer Systems, Prentice-Hall, Inc., 1973.

4. Terplan, K., Reporting on Performance Control Techniques, EUROCOMP Conference Proceedings, p. 379-393, London, 1974.

5. Terplan, K., Reliability and Performance Evaluation of Computing Systems, Information Systems working paper 8-75, University of California, Los Angeles, Graduate School of Management.

6. Smith, A.J., Performance Analysis of Computer Systems Components, Ph.D dissertation, Stanford University, 1974.

7. Gilb, T., Reliable EDP Application Design, Petrocelli Books 1974.

8. Hamming, R.W., Compumetrics: The Way Ahead, INFOTECH State of the Art Report, 18, p. 265-281.

9. Bell, T.E., Computer Performance Analysis: Measurement objectives and tools, Santa Monica RAND (R-584-NASA/PR).

10. Beiser, B., Analytical Preconditions to Simulation, INFOTECH State of the Art Report, 18, p. 477-501.

11. Pinkerton, I.M.M., Practical Criteria for Systems Measurement, INFOTECH State of the Art Report, 18, p. 301-325.

12. Terplan, K., Interaction Between Computing Systems Reliability and Performance Control Techniques, Proc. of the Eighth Hawaiian Information Conference on System Sciences, p. 104-106.

13. Sharpe, W.F., The Economics of Computers, Columbia Univ. Press, New York, 1969.

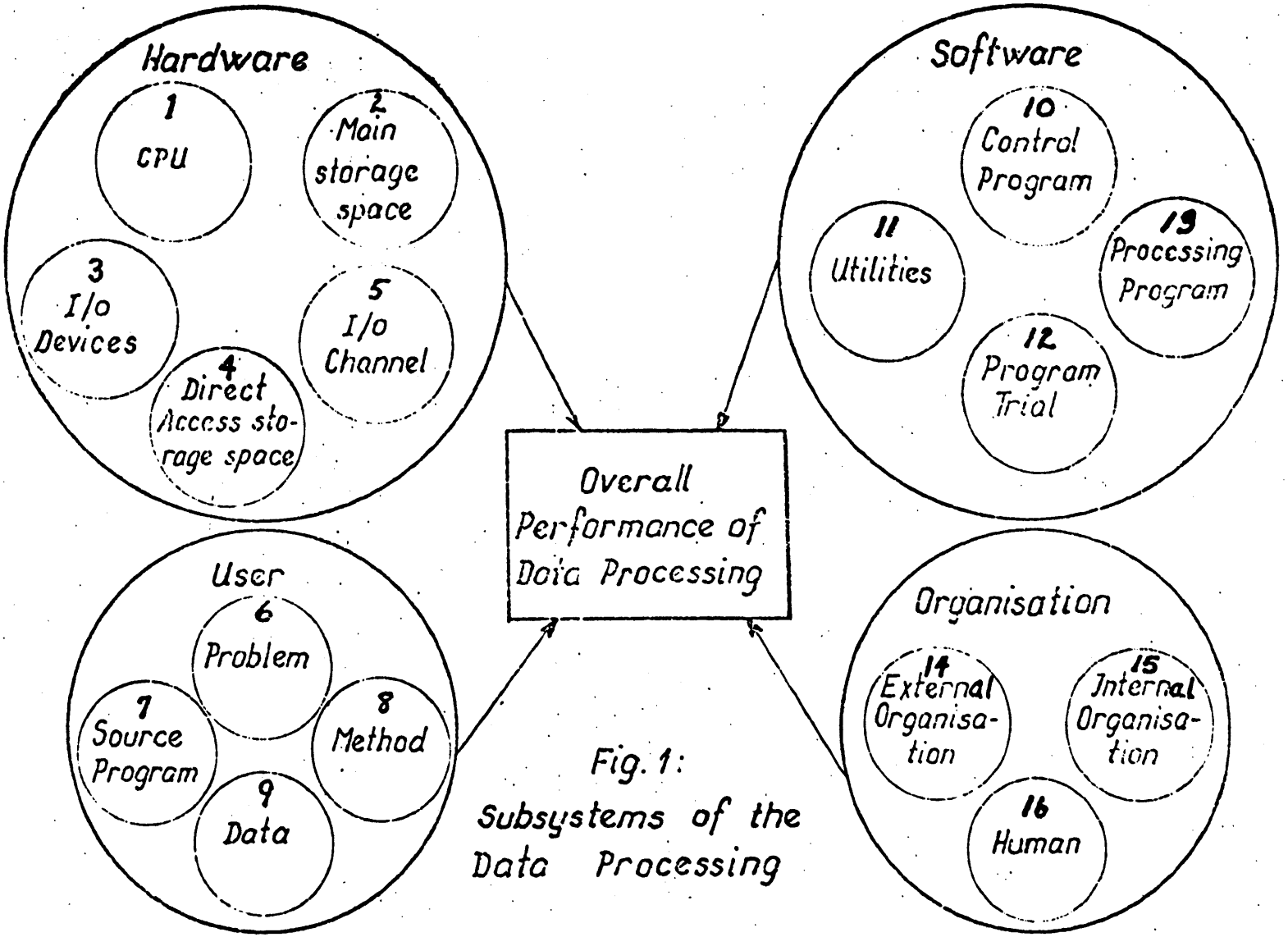

Fig. 1:
Subsystems of the Data Processing