Performance considerations for a distributed data processing system designed for high availability

by S. Agassi

The high-availability requirements of computerized systems that are needed to meet the objectives of the organization are being acknowledged more and more by the data processing community. The paper presents the planning process for a distributed data processing system deisgned to meet high availability requirements. This process was performed as a systems engineering activity in order to assess the feasibility of the presented approach, which was proposed to a customer.

When we are faced with the task of establishing a data processing system, there are many factors that influence the choice of the optimal solution. In a distributed data processing system with distributed data and high-availability requirements, there are many such factors, and their mutual interactions are even more complicated. Among the main questions that we must answer are

- What are the hardware and software configurations (CPU, channels, disks, operating system, data base types) required at the various locations?
- What communication network connections are best suited for the specified requirements?
- What will be the response time of the transactions?
- What will be the effect of the various backup configurations on the response times?
- How will the data base distribution technique affect response times?
- How can data base synchronization be ensured?
- How will the growth in the number of users or in the transaction rate impact the system?

Answers to these questions are vital in assessing the feasibility of a solution, ensuring the optimal investment in system resources, and minimizing risks in the development phases of the system.

The evaluation process presented here was performed iteratively and consumed about four man-months of systems engineering effort, with the full cooperation of the customer. Although this approach was investigated for a specific system, we believe that it is possible, from the presented methodology, processes, ideas, and the conclusions drawn, to develop possible solutions to other distributed systems as well.

Customer requirements

The application. The application is the computerization of a centralized direction and dispatch system.[1] Such systems have two main characteristics:

1. Decision-making based on an up-to-date picture of the situation
2. Communication of operational directives on a real-time basis

An example of such a system is the dispatch system of a taxi company. The control center receives customer calls, including location, time, and desired destination information. Similarly, it receives continuous driver reports regarding taxi location and availability (for-hire/busy). On the basis of this information the control center gives directives to the taxi drivers regarding the times and/or destinations of their trips.

End users must have a highly available system.

The specific system that was investigated deals with the gathering of textual information from various computerized and manual systems, with distribution of the information, in real time, to the various locations, and with its displaying and printing, in different formats, to the various output devices connected with the system.

High-availability requirements. The end users, dealing with real-time decision-making based on the computerized system, must have a highly available system.[2,3] The availability requirements of the system are as follows:

- Central site outages are to be avoided as much as possible.
- Should the central site fail, a remote site will take its place.
- Should the communication among computerized sites fail, each computerized site will be able to work autonomously and continue giving services to the end-user community connected to it.
- Every user will continue to have service after a single computerized-site failure.

Response-time requirements. Four types of response times[4,5] are required:

1. Response to "request for next page": An average of one second or less.
2. Response to "text input" and "simple inquiry": An average of two seconds or less.
3. Response to "data base (DB) update and replication": An average of four seconds or less.
4. Response to "final output distribution": An average of ten seconds or less.

The proposed solution

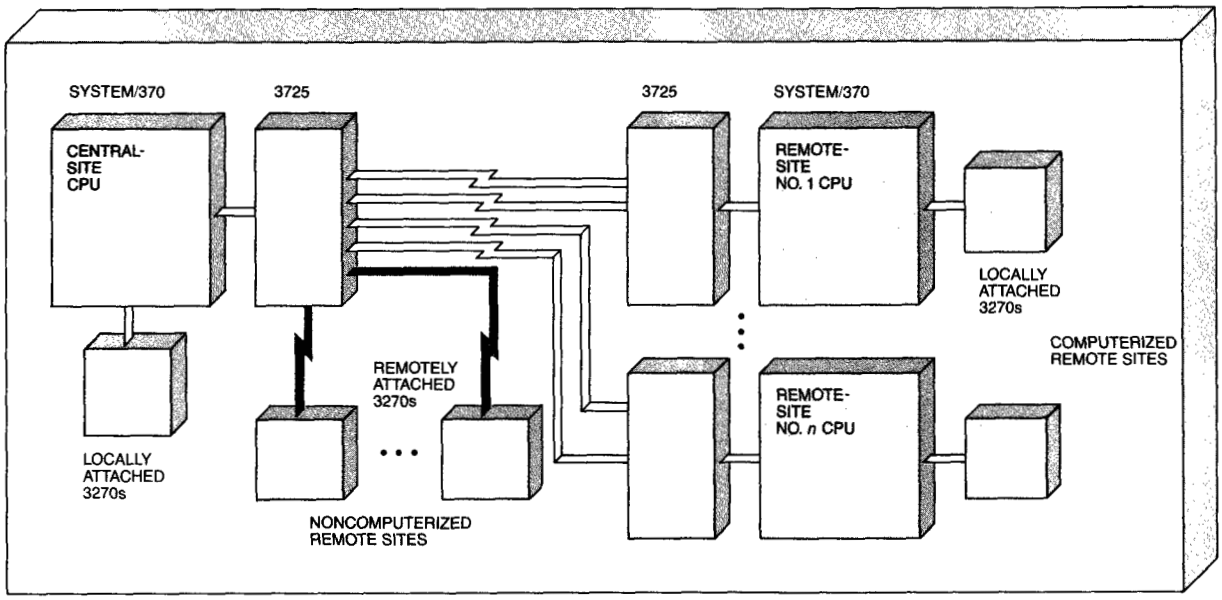
The principal considerations for proposing the solution presented in this paper were well-accepted architectures and products, which are implemented and installed in hundreds of computer installations. These facts are helpful in supporting the feasibility of the solution and in ensuring long-term support, enhancement, and protection of the customer's investment.

Hardware products and network connections. The proposed system is illustrated in Figure 1. It is composed of a central-site computer to which both computerized and noncomputerized remote sites are connected. The computers in the central and remote sites are of the IBM 4381 family.[6] The noncomputerized sites are composed of clusters of the IBM 3270 family workstations.[7] IBM 3725s[8] are used as the communication controllers.

The reasons for choosing a distributed configuration rather than a centralized one are derived directly from the high-availability requirements previously mentioned: By putting computers in several geographically dispersed sites, one can guarantee continuous autonomous service in cases of central site failure or of disconnection of any site from the others.

Additionally, to address the requirement of central site availability, a "hot-standby" CPU is proposed. (See discussion of the "hot-standby" configuration later in the paper.)

Figure 1 Basic configuration of the hardware



To address the high-availability requirements of the network connections, each site is connected physically to at least two computerized sites. Switching to the alternate connection in cases of primary connection failure is done automatically for the interconnections of the computerized sites [exploiting the facilities of alternate routing of Systems Network Architecture (SNA)] and manually for the noncomputerized sites.

Further details about backup implementation are described in the discussion on backup configuration later in the paper.

Software products. The proposed software configuration in the computerized sites is illustrated in Figure 2. It is based on the Disk Operating System/Virtual Storage Extended (DOS/VSE).[9] The data base/data communication subsystem is the Customer Information Control System/Virtual Storage (CICS/VS).[10,11] Communication among the different locations is based on SNA,[12] implemented by the Advanced Communication Function/Virtual Telecommunications Access Method (ACF/VTAM) and Advanced Communication Function/Network Control Program/Virtual Storage (ACF/NCP/VS) program products.[13] Communication among the computerized locations is done via the CICS systems together with the distributed services of the Advanced-Program-to-Program Communication (APPC) architecture[14,15] used by CICS/VS. The data base is the Structured Query Language/Data System (SQL/DS) relational data base.[16] The application programming language is PL/1.[17]

Also proposed is the System/Network Control Center (S/NCC) methodology[18-21] to support the high-availability requirements of the system by an efficient management of the system and network. The principal tools of the S/NCC are the Network Communications Control Facility (NCCF),[22] Network Problem Determination Application (NPDA),[23] and Operator Communication Control Facility (OCCF)[24] program products.

Data base replication. The approach used to support the specific high-availability requirements of the system is to *replicate* the data bases in the central and the remote sites, using CICS/VS SNA APPC (LU 6.2) services. There are two aims to the replication:

1. To enable the autonomous service of a site, in cases of central site failure or disconnection
2. To offer local services to the end users who are connected to the computerized remote sites

It is essential to mention here that the replication in our case had to be executed synchronously (as op-

posed to the deferred-update technique) in order to keep the data as "fresh" as possible, because typically in a centralized direction and dispatch system old data are useless. The technique chosen to accomplish the replication is described later in the paper.

The planning and performance analysis of the solution

The tools. To help in the planning and performance analysis processes in the early stages of the system characterization, we have used the analytic tool based on queuing theory,[25] called ANCICSVS,[26,27] for the analysis of the CICS/VS systems. For the simulation of the communication network we have used the simulation[28] tool SNAP/SHOT.[29]

Application data acquisition. The initial stages of planning and performance analysis of a system demand the definition of the initial data which will describe the communications network configuration, the number and nature of the users connected to it, and the files and transactions which will be processed in the system.

Table 1 gives the "Transaction Profile" defined in our case for each transaction which is executed in the central site. In the table, transactions A to G are the "data base (DB)-update" transactions. DSPLY is a generic transaction and represents the "non-DB-update" transactions, such as display paging and inquiries. PRNT is a transaction that reads from the CICS/VS "transient data" queue file the previously queued records by the DB-update transactions, and sends the record to a printer.

Part of the data was derived directly from the specified requirements. Another part, however, was based on assumptions about the user's working procedures and application solution technique. These assumptions included "pessimistic" (i.e., higher than "normal") loads, in order to offset the possibility of those assumptions not being valid in practice. These assumptions are also the first candidates for verification in the process of the system development.

Even at this early planning stage, it was possible to identify those critical system parameters that drastically influence performance of the system and that require deeper and more detailed investigation.

Figure 2 Software configuration

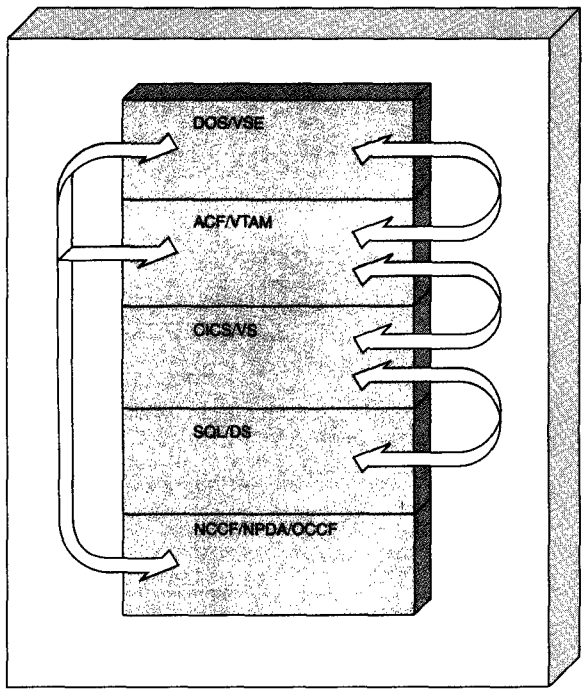


Table 1 Transaction profile

Transaction	Input Message Length (bytes)	Application Path Length (machine instructions)	Data Base Calls (count)	APPC Message Length (bytes)	CICS Transient-Data Writes (count)	Output Message Length (bytes)
A	100	250,000	3	200	4	
B	100	150,000	3	50	2	
C	0	100,000	3	50	6	
D	500	250,000	3	1000	4	
E	100	250,000	3	100	8	
F	1500	500,000	8	2000	4	2000
G	1000	500,000	8	2000	4	
DSPLY	20	60,000	1	0	0	1000
PRNT	0	0	0	0	1 (Read)	740

Figure 3 A duplicate data base update CICS/VS transaction

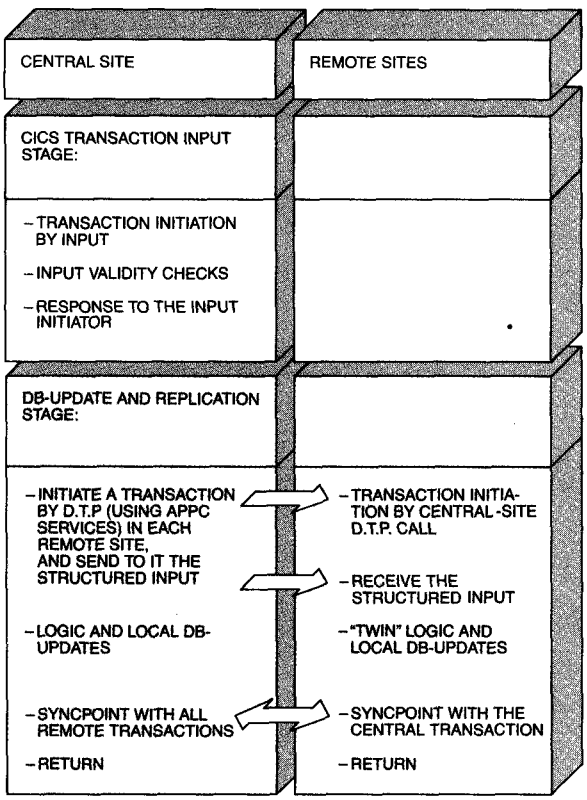


Table 2 Transaction arrival rates in the various load scenarios (in transactions per minute)

Transaction	Peak		
	No. 1	No. 2	No. 3
A	6.5	9.0	6.0
B	29.8	0	27.0
C	3.0	3.0	2.5
D	0	9.5	0
E	0	38.0	19.0
F	9.0	0	7.0
G	6.0	4.0	5.0
DSPLY	259.0	259.0	259.0
PRNT	191.0	413.0	317.0

The various transactions were described earlier under "Application data acquisition."

Transaction rates. There is a close relationship among the resources to be invested in the system, the response times required from the system, and its transaction rates. There is, therefore, great importance attached to the transaction rate parameter. On one hand, if the parameter is defined too high, we will have a system in which the resources are underutilized and the investment wasted. On the other hand, if it is defined too low, we will find a system which has over-utilized resources and consequently offers an inferior service to its users.

Normally the system is designed to give an optimal level of service to its users according to the average transaction load during the daily peak hour. Whether there is any benefit in investing for loads that occur once every week or every month depends on the nature of the system and on how critical the quality of its service is during those peak periods. In such systems it is possible to achieve economic recompense by the use of system resources by other, noncritical, services (such as program development) during nonpeak periods. Those services must be able to be switched quickly from the system as necessary.

Table 2 gives the various load scenarios that were defined in our case. Each scenario is composed of a specific transaction mix which represents the predicted occurrence of processes in the real system.

Choice of the data base distribution technique. One of the goals set during the planning phase was to analyze the various available distribution techniques and to recommend the preferred one. The CICS/VS system provides two main techniques to achieve distribution.

1. The first is "Function Shipping," which is a distribution of a single request—in our particular case, a single data base access. In this technique, transactions being processed in the central site can update both the local data bases and those located in each of the remote sites, as if they were all local.
2. The second is "Distributed Transaction Processing" (DTP), in which the distribution is controlled by a dialogue between remote transactions. This technique can be used in our case, as illustrated in Figure 3. A DB-update transaction, initiated in the central site, initiates an identical transaction in each remote site. As a result, the data bases in each remote site are updated in the same manner as in the central site.

In the system under investigation we found that the Function Shipping technique demanded more resources of the central site and the communication lines, while the Distributed Transaction Processing technique loaded the remote sites more heavily but gave better response times.

Table 3 Time spent during execution of the transactions (seconds)

Transaction	Basic Configuration Total Response Time	CPU/Line Time	I/O Time	APPC Time
A	3.62	1.20	0.29	2.13
B	3.97	1.89	0.28	1.80
C	3.17	1.05	0.31	1.80
E	4.27	1.96	0.32	1.99
F	8.92	2.44	0.66	5.81
G	8.86	2.44	0.64	5.78
Local DSPLY	0.21	0.18	0.03	0.0
Remote DSPLY	2.20	2.17	0.03	0.0
Local PRNT	0.10	0.10	0.0	0.0
Remote PRNT	1.65	1.65	0.0	0.0

The response times are for Peak No. 3, which was found to be the "heaviest."
The various time definitions are as follows:

- CPU/line time: Time in seconds that the transaction spends processing in the CPU, waiting for the CPU following completion of I/O accesses, transmitting, and waiting for the line.
- I/O time: Time in seconds that the transaction waits for completion of I/O accesses.
- APPC time: Time in seconds that the transaction waits for completion of remote requests via the APPC facility, including session wait, link transmission, and the *called* response times of transactions.

(The presented response times are for accessing VSAM files and not SQL/DS data bases, since ANCICSVS currently does not support SQL/DS analysis.)

From an analysis of the results the conclusion drawn was that choosing the Distributed Transaction Processing technique would give the system the best performance.

Thorough consideration should be given to the data base synchronization aspect. What happens if the central data base is updated and one of the remote data bases is not because of failure before the remote site gets the updating transaction? What happens if the remote transaction fails before synchronizing with the central transaction? (Have the remote data bases already been updated or not?) Answers to these kinds of questions must be found in order to ensure the synchronization of the data bases. In the system under investigation we found that the syncpoint mechanisms of CICS/VS, SQL/DS, and APPC placed together[32,33] are very helpful in identifying and handling such situations.

Establishment of the basic system configuration. After the initial definition of the system configuration, transaction profiles, load scenarios, and the data base distribution technique, initial test runs of the analytical models, using ANCICSVS, were done. During these runs, an iterative system tuning was performed. It was found that a combination of IBM 4381 Model Group 1 CPUs and 9600-bit-per-second lines would give acceptable performance. Table 3 presents the response times of the various transactions. The presented response time is the average elapsed time between hitting the "Enter" key that initiates the message transmission from the workstation to the host and the return of the last character of the output message to the originating workstation. It includes CPU plus line time, I/O time, and APPC time consumed by the execution of the transaction.

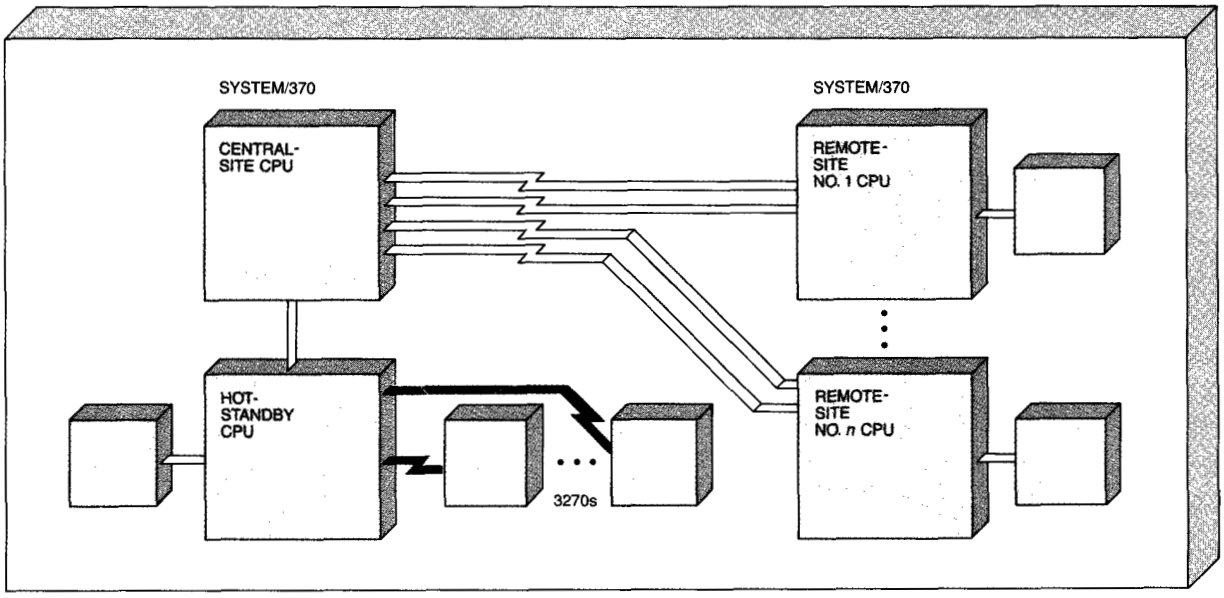
Table 4 Response time comparison

Response Type	Response Times (sec)	
	Required	Achieved (weighted average)
"request for next page"	≦1	Local: 0.21 Remote: 2.20
"text input"	≦2	1.92
"simple inquiry"	≦2	Local: 0.21 Remote: 2.20
"DB-update and replication"	≦4	4.88
"final output distribution"	≦10	Local: 4.98+ Remote: 6.53+

Table 4 provides a comparison between the required and the achieved response times.

The derivation of the estimated response times is done, using the figures in Tables 2 and 3, as follows:

Figure 4 Hardware configuration changes – adding a "hot-standby" configuration



- Response to "Request for Next Page" and "Simple Inquiry" equals the DSPLY transaction response time.

- Response to "Text Input" equals (approximately) the weighted average of the DB-update transactions *total* response time minus the sum of its APPC time and I/O time.

  This calculation needs some more clarification: From Figure 3 one can see that the response to the input-initiator is sent *after* the input-validity checks and *before* the data base replications (which consume principally the APPC time) and the data base updates (which consume principally the I/O time).

  Now, the weighted average of the DB-update response time of the transactions is calculated as follows:

$$RT = \frac{\sum_i (AR_i \times RT_i)}{\sum_i (AR_i)}$$

  where AR_i and RT_i are the arrival rate and the response time of transaction i, respectively, where $i = A, B, \cdots, G$.

  Numerically, the response time is

$$RT = \frac{(6.0 \times 3.62 + 27.0 \times 3.97 + \cdots + 5.0 \times 8.86)}{(6.0 + 27.0 + \cdots + 5.0)}$$
$$= 4.88 \text{ seconds}$$

  In a similar way, one can find that the weighted average of the APPC time and I/O time of the DB-update transactions are 2.60 and 0.36 seconds, respectively, giving the response for "Text Input" of $4.88 - (2.60 + 0.36) = 1.92$ seconds.

- Response to "DB-update and replication" equals the weighted average of the response time of the DB-update transactions, calculated above.

- Response to "Final Output Distribution" equals the weighted average of the response time of the DB-update transactions plus PRNT transaction INIT time plus PRNT transaction response time.

It was concluded from the results that line speeds greater than 9600 bits per second are required in order to achieve all the required response times.

Table 5, column 1 gives the utilization levels of the various system resources (CPUs, channels, and communication lines). The utilization figures permitted, apart from determination of the quantity of required resources, an estimation of the highly loaded re-

Table 5 Resource utilization (in percent) in the various configurations

Resource	Configuration						
	No. 1 (Col. 1)	No. 2 (Col. 2)		No. 3 (Col. 3)		No. 4 (Col. 4)	
		Central CPU	Hot-Standby CPU	Central CPU	Hot-Standby CPU	Central CPU	Hot-Standby CPU
CPU	71.9	51.9	50.8	55.8	58.0	68.8	67.2
CHAN 1	13.0	7.0	13.0	7.0	15.0	10.0	18.0
CHAN 2	13.0	13.0	1.0	18.0	1.0	17.0	2.0
TG-Link	7.9	7.9	0.0	7.9	0.0	10.4	0.0
LNGRP 1	26.0	15.0 P	12.0 D	15.0 P	12.0 D	20.0 P	17.0 D
LNGRP 2	13.0	8.0 P	6.0 D	8.0 P	6.0 D	10.0 P	8.0 D
LNGRP 3	18.0	0.0	18.0 D	0.0	18.0 D	0.0	24.0 D
LNGRP 4	5.0	0.0	5.0 D	0.0	5.0 D	0.0	6.0 D
LNGRP 5	2.0	0.0	2.0 D	0.0	2.0 D	0.0	2.0 D
LNGRP 6	0.0	0.0	0.0	80.0 P	23.0 D	0.0	0.0
LNGRP 7	0.0	0.0	0.0	80.0 P	20.0 D	0.0	0.0

The various configurations are as follows:

- Configuration No. 1 is the "basic configuration" (see Figure 1).
- Configuration No. 2 is the "hot-standby configuration" (see Figure 4).
- Configuration No. 3 is the "backup configuration" (see Figures 5 and 6).
- Configuration No. 4 is the "unexpected peak growth of 33 percent configuration," which is similar to the hardware configuration of No. 2.

The CPUs are IBM 4381 Model Group 1. The line speed is 9600 bits per second.
"D" and "P" stand for display output and printing output distribution, respectively.
"TG-Link" stands for "transmission group" link, which is the group of lines that interconnect two adjacent IBM 3725 Communication Controllers.
"LNGRP" stands for line group.

sources on which to focus attention during the planning and development of the system, and permitted the estimation of the "margin for error."

"Hot-standby" configuration. At this stage in the planning process, an additional computer was introduced into the central-site system. This computer was designed to be the "hot-standby" computer, which would receive the logical connections (SNA sessions) to the remote sites in the event of central computer failure (see Figure 4). The "hot-standby" CPU improves the availability of the system by reducing the time needed to recover from central CPU failures. This "hot-standby" technique requires the continuous replication of the central computer data bases to the backup computer, in a way similar to the remote sites. (The replication is needed in spite of the fact that the data bases are local to the "hot-standby" system, because it is not currently possible to switch the CICS/VS log file automatically between the two systems.) The concept explored in this situation was "workload sharing" between the central CPU and the "hot-standby" CPU during normal operation (i.e., while *not* in a backup mode of operation).

The tested workload sharing was of the output distribution task from the central site to the various end users. The central CPU would handle the output distribution to the display workstations, while the "hot-standby" CPU would do it for the printers. This sharing is done by taking advantage of the basic SNA facility that enables the *logical* connections between the various workstations and the various applications to be independent of the *physical* network configuration. Table 6, column 2 shows that by adding the "hot-standby" CPU and splitting the distribution task between the two CPUs in the central site, a response time increase is expected for the DB-update transactions due to the additional node which needs data base replication. The "non-update" transactions would have lower response times, and an improved capability of growth would be gained by lowering the utilization of the CPUs. (See Table 5, column 2.)

Backup configuration—Performance and procedures. The next stage in the planning process was to examine the central system ability to give user service in the event of failure of a computerized remote site (see Figure 5). Two aspects were examined:

- The effect on the total system derived from the additional load put on the central site because of the connection of the backed-up workstation to it
- The best network connection of the backed-up workstations

Figure 5 Hardware backup configuration

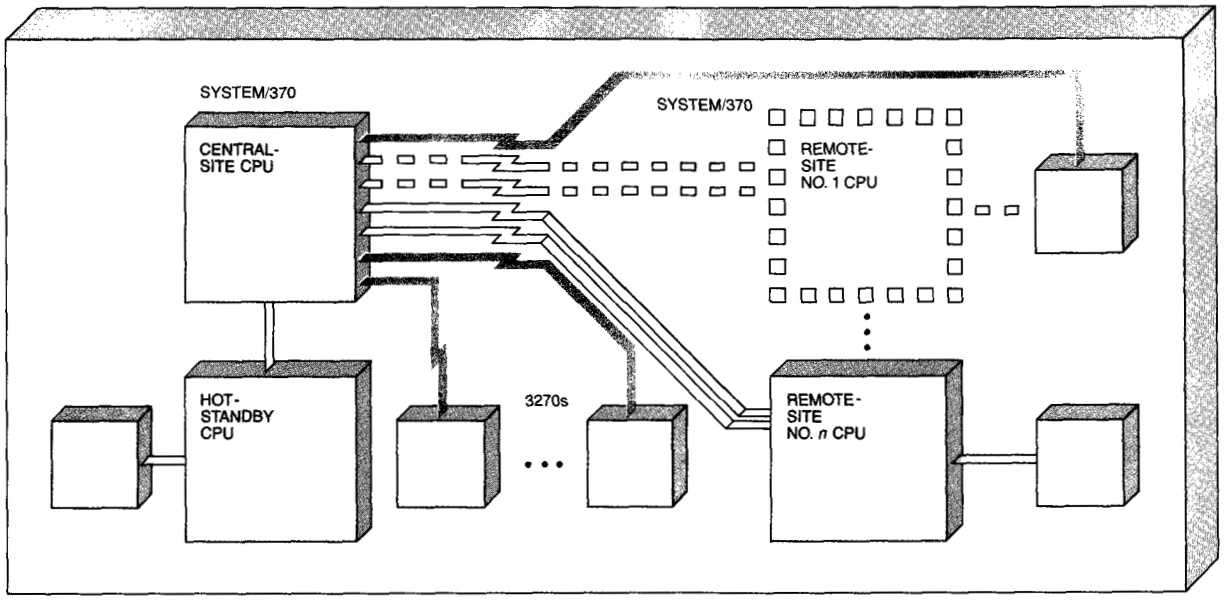


Table 6 Transaction average total response times in the various configurations (seconds)

Transaction	Configuration			
	No. 1 (Col. 1)	No. 2 (Col. 2)	No. 3 (Col. 3)	No. 4 (Col. 4)
A	3.62	4.06	2.80	5.08
B	3.97	4.35	4.00	5.84
C	3.17	3.49	2.43	4.53
E	4.27	5.27	4.08	6.44
F	8.92	9.91	6.59	11.97
G	8.86	10.01	6.73	12.10
Local DSPLY	0.21	0.15	0.16	0.18
Remote DSPLY	2.20	2.20	2.20	2.20
Local PRNT	0.10	0.09	0.07	0.11
Remote PRNT	1.65	1.30	1.58	1.49

The various configurations are defined under Table 5.

An analysis of the results showed that the central-system configuration, chosen for normal operations, would also provide the required user service in backup configuration. The additional load on the central site would be absorbed without impacting the service level beyond the planned objective level. (See Table 5, column 3 and Table 6, column 3.) The response time gain for the DB-update transactions is due to the fact that in this configuration there is one less node that needs data base replication. At this point one can see that high availability is not gained "free" but may cost in response time!

Figure 6 illustrates the different network connections of the backed-up workstations examined: link up to the central site either directly or via IBM 3725 communication controllers located at the remote site.

It was concluded, using SNAP/SHOT, that the linking of remote workstations to the center via communications controllers can give better service to users than a direct link. The service quality under this topology depends on the number of parallel trunk lines connected between the central and the remote communications controllers and also on the transmission block sizes (RU size), thus permitting parallel data transmission. A number of simulations were made by which the required number of trunk lines and the optimal message lengths that yield the best response times were determined.

Let us take a short glimpse at the proposed solution for the switching of the terminals from the normal to the backup mode of operation.

- Standby 3274s are kept directly connected by 57 600-bit-per-second links to the communications controller in the remote site for the backup purpose.
- The IBM 3270 workstations are attached by coaxial cables through a manual switching board to both the local 3274s (which are operational during the normal mode of operation), and the "remote" 3274s (which are operational during the backup mode of operation).
- Switching between the normal and backup modes of operation and vice versa is performed manually, using the switching board.

In order to restore a failed site to service, it is necessary to synchronize its data base with the other data bases in the system. The principal ideas that were considered to accomplish this task in the system under investigation are as follows:

- Delay the restoring process to a proper *nonpeak* period during which the number of data base updating transactions is small. This fact minimizes the impact of failure and the degradation of the system service level during the restoring process.
- Transmit a full fresh copy of the data base from the center to the restored site. This is acceptable due to the fact that the data base is relatively small.
- Hold the data base updating transactions which arrive during the restoring process, if any, in the central site and delay their processing until the completion of the restoring process.
- When the restoring process is completed, process the data base updating transactions that were previously withheld from processing and, by this, complete the data base synchronization.

Planning for growth. An additional topic examined was the influence of a 33 percent load growth on the system. This growth could result either from unexpected peaks or from planned gradual growth in the number of users.

To check the effect of an unexpected peak, one observes the response times achieved, using resources of the system that were designed for operation in the originally specified load. Table 5, column 4 and Table 6, column 4 show that system saturation was *not* expected to occur, and that system response times would be degraded by one to two seconds compared with those in the originally specified load. Response time increase is due to the global increase of the utilization of the resources in the whole system, a fact that causes longer queuing times.

Figure 6 Hardware backup topologies linking backed-up 3270s

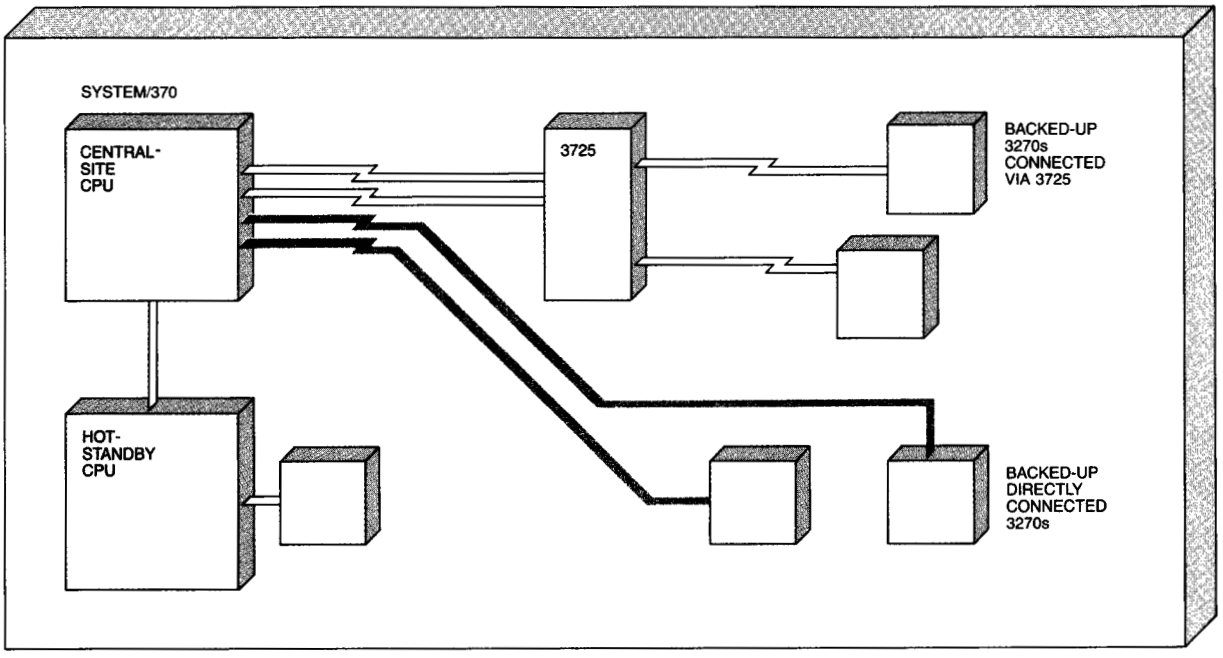


Table 7 Response time reductions (seconds) due to enhancement of the resources of the system

Transaction	Configuration Change			
	No. 1	No. 2	No. 3	No. 4
A	0.94	0.46	0.30	0.0
B	0.89	0.41	0.13	0.16
C	0.90	0.43	0.13	0.0
E	1.11	0.48	0.19	0.15
F	1.96	0.93	2.40	0.0
G	1.96	0.91	2.40	0.0

The various configuration changes were described earlier under "Planning for growth."

The check on the effect of gradual growth was performed on an expanded system configuration, in order to understand which system resources (CPU, lines) would optimally require enhancement to maintain the same response times as those obtained at the originally specified load. In our case, several simulation runs were made in which various configuration changes and their effects on response times were examined. Table 7 gives the results of the various response time reductions achieved due to the following configuration changes:

- Configuration Change No. 1: Upgrading the central-site CPUs from 4381 Model Group 1 to Model Group 2.
- Configuration Change No. 2: Upgrading the remote-site CPUs from 4381 Model Group 1 to Model Group 2.
- Configuration Change No. 3: Changing the TG-links from 10 × 9600-bit-per-second to 2 × 32 000-bit-per-second links.
- Configuration Change No. 4: Adding lines to a specific noncomputerized site.

From an analysis of the results, it can be seen that by enhancing only the central CPUs and the links between the central and the computerized remote sites (Configuration Changes No. 1 and No. 3 together), the desired objective, i.e., to have response times the same as or better than in the originally specified load, would be achieved (except for Transaction B). (The calculation is done by subtracting the sum of columns 1 and 3 in Table 7 from column 4 of Table 6, and comparing it against column 2 of Table 6.)

One can argue at this point that multiple changes may not necessarily provide additive results on response time improvement. In terms of statistics, in many real-life applications, the assumptions of Poisson distribution of the message arrival rate and exponential distribution of the message service are sufficiently adequate representations of real data. In such systems, Jackson's Theorem[25] holds, justifying the assumption that mean response time of each component in the system (in our case CPUs and communication lines) may be added to derive the total system response time.

Additional system planning considerations

In order to maintain as close a correlation as possible between the performance estimates and actual performance, the estimating process must be repeated during system development each time a significant change is made that was not taken into account in previous estimates.

In the later stages of system development, more sophisticated planning tools may be used to generate transactions from a simulated communications network to the real system. In this way we can analyze how the tested system is functioning under stress conditions and measure its performance. The tool used for this purpose in IBM is the Teleprocessing Network Simulator (TPNS).[30,31]

Apart from the distributed system performance analysis, there are, of course, other topics which must be addressed in the planning phase. Among them are

- The recovery and restart processes[32,33] after failure
- The S/NCC implementation
- The human-factors[34] attitudes and features concerning the man/machine interface (colors, menus, windows, PF-keys, light-pen, etc.)
- The incorporation of other current and future systems
- The data base design

A discussion of these topics is beyond the scope of this paper.

Concluding remarks

This paper has presented a description of the planning process for a distributed data base and data processing system, designed to meet high-availability requirements.

The use of planning tools permitted the proposal of the optimal system configuration for normal and backup mode operation, determined the critical areas which require further, deeper investigation, indicated the most efficient data distribution tech-

nique, and facilitated an estimation of the system growth capabilities. A direct consequence of this was the validation of the proposed approach to achieve the specific requirements of the system. Also, it enabled us to acquire reliable information that could help the project manager make decisions about the optimal system solution to be offered to his users. This information can contribute to supporting the cost analysis, to efficient system development, and to reduction of development risks.

Acknowledgments

The author wishes to gratefully acknowledge Raviv Karnieli, who took an active role in the planning and the analysis process, encouraged the writing of this paper, and also gave valuable comments on the draft copy. The author also wants to express a special acknowledgment to Mike Wilmot from IBM Israel, who translated the draft copy of the paper from Hebrew to English. In attending a Residency Project on the subject of "high availability" which took place in the Raleigh International Systems Center, Raleigh, North Carolina, the author was fortunate to receive valuable comments on the paper from the following IBM systems engineers: Guy Le-Roy (Raleigh ISC), Norbert Scheer (IBM Germany), and Frans Jansen (IBM Netherlands). Emanuel Gruengard, IBM Israel's Country Systems Engineering Manager, gave valuable comments which helped the author to bring the paper to its completion.

Cited references

1. D. J. Morris, *Communication for Command and Control Systems*, International Series on Systems and Control, Vol. 5, Pergamon Press, Elmsford, NY (1983).
2. *Winning Availability Strategies Videotape Transcript*, GG22-9372, IBM Corporation (June 1984); available through IBM branch offices.
3. *Systems Analysis for High Availability, An Availability Management Technique*, GG22-9391, IBM Corporation (January 1985); available through IBM branch offices.
4. R. B. Miller, "Response time in man-computer conversational transactions," *AFIPS Conference Proceedings, Fall Joint Computer Conference* **33**, Part 1, 267–277 (1968).
5. W. J. Doherty and A. J. Thadhani, *The Economic Value of Rapid Response Time*, GE20-0752, IBM Corporation (November 1982); available through IBM branch offices.
6. *4381 Processors Model Groups 1 & 2 Functional Characteristics*, GA24-3947, IBM Corporation (March 1984); available through IBM branch offices.
7. *An Introduction to the IBM 3270 Information Display System*, GA27-2739, IBM Corporation (December 1983); available through IBM branch offices.
8. *Introduction to the IBM 3725 Communications Controller*, GA33-0010, IBM Corporation (October 1984); available through IBM branch offices.
9. *VSE/SP 2.1 General Information*, GC33-6176, IBM Corporation (June 1984); available through IBM branch offices.
10. *Customer Information Control System/Virtual Storage (CICS/VS) General Information*, GC33-0155, IBM Corporation (August 1983); available through IBM branch offices.
11. *IBM CICS/VS Direction and Strategy*, G320-5890, IBM Corporation (January 1985); available through IBM branch offices.
12. *Systems Network Architecture (SNA), Concepts and Products*, GC30-3072, IBM Corporation (February 1984); available through IBM branch offices.
13. *Network Program Products General Information Manual*, GC27-0657, IBM Corporation (November 1983); available through IBM branch offices.
14. *An Introduction to Advanced Program-to-Program Communication (APPC)*, GG24-1584, IBM Corporation (July 1983); available through IBM branch offices.
15. J. P. Gray et al., "Advanced program-to-program communication in SNA," *IBM Systems Journal* **22**, No. 4, 298–318 (1983).
16. *SQL/Data System Concepts and Facilities for VSE*, GH24-5013, IBM Corporation (November 1984); available through IBM branch offices.
17. *OS and DOS PL/I Language Reference Manual*, GC26-3977, IBM Corporation (September 1981); available through IBM branch offices.
18. *System/Network Control Center Overview—Processes and Products*, G226-3551, IBM Corporation (June 1983); available through IBM branch offices.
19. *Developing a Multi-Site Network Management Plan*, GG22-9365, IBM Corporation (July 1984); available through IBM branch offices.
20. R. D. Garrigues, "An application of network management at a large computing service," *IBM Systems Journal* **22**, Nos. 1/2, 143–164 (1983).
21. T. P. Sullivan, "Communications Network Management enhancements for SNA networks: An overview," *IBM Systems Journal* **22**, Nos. 1/2, 129–142 (1983).
22. *NCCF V2 R2, Installation and Functions Guide*, GG24-1666, IBM Corporation (October 1984); available through IBM branch offices.
23. *Network Problem Determination Application (NPDA) General Information, Version 3*, GC34-2111, IBM Corporation (June 1983); available through IBM branch offices.
24. *VSE/Operator Communication Control Facility (OCCF) General Information*, GC33-6113, IBM Corporation (July 1980); available through IBM branch offices.
25. *Analysis of Some Queuing Models in Real-Time Systems*, GF20-0007, IBM Corporation (September 1971); available through IBM branch offices.
26. P. H. Seaman, "Modeling considerations for predicting performance of CICS/VS systems," *IBM Systems Journal* **19**, No. 1, 68–80 (1980).
27. R. D. Acker and P. H. Seaman, "Modeling distributed processing across multiple CICS/VS sites," *IBM Systems Journal* **21**, No. 4, 471–489 (1982).
28. G. S. Fishman, *Concepts and Methods in Discrete Event Digital Simulation*, Wiley Interscience, New York (1973).
29. H. M. Stewart, "Performance analysis of complex communications systems," *IBM Systems Journal* **18**, No. 3, 356–373 (1979).
30. *Teleprocessing Network Simulator (TPNS) General Information*, GH20-2487, IBM Corporation (September 1984); available through IBM branch offices.
31. *TPNS Version 2 Release 3 Highlights, Network and Application Development Testing*, GG22-9384, IBM Corporation (September 1984); available through IBM branch offices.

32. *CICS/VS 1.6 Recovery and Restart Guide*, SC33-0135, IBM Corporation (June 1983); available through IBM branch offices.
33. *CICS/VS 1.6 Intercommunication Facilities Guide*, Part 5, SC33-0133, IBM Corporation (June 1983); available through IBM branch offices.
34. *Human Factors of Workstations with Display Terminals*, G320-6102, IBM Corporation (September 1979); available through IBM branch offices.

Shimon Agassi *IBM Israel Limited, P.O. Box 33666, Tel-Aviv, Israel 61330.* Mr. Agassi joined the IBM Corporation in 1971. Since that time he has been a systems engineer in a branch office, where he has been involved in a variety of systems engineering activities in the area of teleprocessing: developing applications, planning networks, installing products for networks and for DB/DC and distributed processing systems, planning and implementing migrations from non-SNA to SNA networks, including interconnections of standard and nonstandard networks and teleprocessing products. Most of these activities were performed for large-systems accounts. Since 1978 he has been a teleprocessing branch specialist. Mr. Agassi has a B.S. degree in statistics and economics from Tel-Aviv University.

Reprint Order No. G321-5249.