DISTRIBUTED PROCESSING USING IMF/3000

by Michael E. Ura

Over the past five years, much has been written about distributed processing. The concepts are exciting. The ability for a user to sit down at a terminal to access and process data no matter where it resides is certainly the key to success in building a distributed system. Unfortunately, translating distributed concepts into real-world, cost-effective application systems is very difficult. Moreover, vendors have only been able to supply the "tools" for building distributed systems. Very little software is on the market to support true distributed system technology. This paper will address the use of several HP tools, most significantly IMF/3000 (Interactive Mainframe Facility) to create a distributed Inventory Management System utilizing large- scale IBM mainframes (IBM 30xx) and HP/3000 series computers.

To bring the paper into perspective, a brief description of the company's environment is necessary. Monsanto is a large ($6 billion plus in sales) chemical manufacturer consisting of five operating companies each with plants located throughout the United States. Each operating company is responsible for formulating, within reason and corporate guidelines, a hardware and system strategy. The Monsanto Industrial Chemicals Company (MIC) has chosen the HP/3000 Series of computers and its associated software to support Information System Processing at headquarters and each plant location. Figure 1 is a layout of the present network. These HP computers provide each plant location the ability to develop and process in-plant systems independent of headquarters while continuing to feed and access companywide systems when necessary. Implementation of the HP/3000 strategy within MIC began in 1981 and will be completed in 1984.

As with most companies, inventory control is a key function. The responsibility for inventory control in Monsanto is delegated to each operating company with the only corporate requirements being the reporting of monthly inventory balances for inclusion in the General Ledger System. Figure 2 outlines MIC company's inventory control function. As can be seen, inventory information and reporting requirements exist throughout the organization, both in a central location and at each plant location. Because of this central need, and the state of computer technology in 1977, the decision was made to design a large inventory data base system on the IBM mainframe, utilizing IBM's product IMS (Information Management System). The system was essentially batch oriented with no on-line support facilities, even though the IMS software was fully capable of handling on-line communications. Plants were required to use the system to report and manage their inventories. While the system served the headquarters inventory needs very well, it was incapable in its batch mode to keep up with the daily inventory needs of the plants -- the most important inventory control point. As MIC entered the economic recession of 1981 the need for better inventory controls across Monsanto became evident and the MIC plants cried out for help. The question at that point was, "Do we scrap our present Inventory Control System, a $1.3M investment, and utilize our now emerging HP/3000 strategy?" or "Do we attempt to build on our Inventory Control System and fit the HP/3000 in somehow?"

This was not an easy decision to make. Some pertinent issues concerning the predicament will make the transition to the use of the HP/3000 in a distributed mode more evident.

1. The present Inventory Control System was basically good. Its data base was well designed and its programs were virtually error-free. All company and corporate interfaces already existed and worked well.

2. The inventory information needs of the users at headquarters were being met.

3. The inventory information needs of the plants were not being met. The batch mode of processing was too slow and costly to meet the demands of the plants.

4. Each plant operated differently. Their inventory information needs as well as the way they processed inventory information was different. In some cases the only common element was that they received raw materials, produced products and shipped the products to customers.

5. Having an HP/3000 strategy approved, it was not feasible to recommend any other vendor.

Issue four was the key to the decision to build a distributed system. A major requirement put forth by the plants was that any system must meet the needs of the plant; therefore, a common solution was not acceptable. But how to fit a centralized system into a plant-unique environment is not an easy question to answer, especially in an existing system so dominated by IBM technology. The key to the solution was HP's Interactive Mainframe Facility (IMF/3000).

HOW THE SYSTEM WORKS

IMF/3000 operates in two modes, pass-thru and programmatic. Pass-thru simply makes an HP26xx terminal look like an IBM 3270- type terminal. A communications line from the IBM mainframe is attached to an HP/3000 with IMF/3000 installed. IMF makes the HP/3000 look like an IBM 3274 control unit, giving all HP26xx terminals the ability to look like an IBM 3270 terminal (CRT) device. This facility allows a user to not only access the HP/3000 but also allows him to access facilities on the IBM mainframe such as TSO, CMS and IMS application systems. The end result is that a user does not have to have two CRT terminals on his desk if he requires access to both the HP and the IBM. Figure 3 compares a pass-thru environment with an IBM 3270 environment.

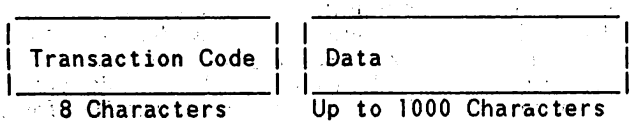
IMF/3000 programmatic access emulates an IBM 3270 environment much the same way except that a program is written, usually COBOL, to control inputs and outputs of the user in conjunction with inputs and outputs of the IBM mainframe. With the ability to control both the HP and the IBM communications the problem of fitting plant-unique processing into a centralized system was neatly solved.

The end result was the creation of a multi-plant inventory system with a centralized data base residing on an IBM mainframe with distributed processing of that data by multiple site HP/3000's in an on-line mode. Access to the data base is provided using the IMF/3000 programmatic links to IBM's program product IMS (Information Management System) with screen formatting provided on the HP via VIEW/3000 and resident HP files using KSAM/3000. Figure 4 provides a pictorial view of the system and its files.

As can be seen in figure 4 there are essentially four parts to the system: IMS, IMF/3000, VIEW/3000 and the preprocessor. Each of these will be discussed in detail then will be put together for a general discussion of how the parts interact as a whole.

IBM'S INFORMATION MANAGEMENT SYSTEM (IMS)

IMS, as typical of most IBM products, is very complex. It provides for the creation and maintenance of data bases; batch and on-line access to those data bases and provides a language (DLI - Data Language I) for accessing data bases from COBOL, PLI and Assembler language programs. Two concepts are key to understanding the use of IMS in the distributed system. First, the Inventory Data Bases are IMS data bases and therefore must be accessed by programs on the mainframe normally referred to as IMS programs. In the Inventory Control System these programs are COBOL programs that access the data bases using subroutines (or technically DLI calls). These subroutine calls are very similar to IMAGE data base intrinsics on the HP/3000. The second concept is that of on-line program execution. Programs that are to execute in an on-line mode reside on special program libraries on the IBM defined to IMS. Each program is assigned an eight-character transaction code. This transaction code identifies not only the program to be executed but also identifies which data bases are to be accessed. When IMS receives an input message over the communications line -- a message looks like this:



-- the transaction code is interrogated and the proper program is executed. The program then simply acts on the data received and returns a new message to IMS which in turn sends the outputted message back to the same source as the inputted message. This source is normally a user sitting at a terminal but in our case is the IMF/3000 software on the HP/3000. Under the distributed system the appropriate message is simply sent to IMS and received from IMS via the IMF/3000 software. Program execution including queue time is normally less than three seconds with overall response time in the 5-7 second range dependent on communication line speed and volume of data being passed on the line.

HP'S INTERACTIVE MAINFRAME FACILITY (IMF/3000)

IMF/3000, as mentioned earlier, provides a means for controlling communications (data to be sent across the line) between an HP application program and an IBM mainframe. An application program using IMF looks to the IBM as if it is a 3270 CRT terminal; therefore, the application program must tell the IBM what it is doing by calling IMF intrinsics. The majority of the intrinsics are fairly simple to use. The following is a list of the most commonly used intrinsics:

OPEN3270	Open communication between the application program and the communication line
STREAM3270	Puts data from the application program into the IMF buffer
TRAN3270	Transmits the contents of the IMF buffer to the IBM host
RECV3270	Receives data from the IBM host and puts it in the IMF buffer
READSCREEN	Gets data from the IMF buffer for the application program to use
CLOSE3270	Terminates communications between the application program and the communication line

There is also an intrinsic, ACQUIRE3270, that actually allows you to start an IMF pass-thru session from your program with the program regaining control upon exiting of pass-thru mode.

While these intrinsics look simple and are easy to code, talking to an IBM host via IMF is not easy. The key is knowing exactly how the IBM will respond in any given situation. For example, the application program may send an IMS message to the IBM upon execution of a TRAN3270 intrinsic. The program will then immediately execute a RECV3270 which waits for a response from the IBM host. When a response is received the program executes a READSCREEN and, Guess what? You didn't get what you expected. The reason for this is that sometimes, but not always, the IBM will simply send an acknowledgement that it received the data sent. In this case the first message is discarded with a second RECV3270 and READSCREEN returning the expected response. In other cases a system broadcast message such as "THE COMPUTER IS GOING DOWN IN 5 MINUTES" may be received. Obviously the user needs to see this message; therefore, it cannot be discarded. Again, a second series of RECV3270 and READSCREENS will result in the proper data being received from the IBM host.

This is just one example of the problem in using IMF intrinsics. For those familiar with IBM on-line systems the intrinsics work excellently with IMS but very poorly with TSO and CMS. The reason for this is that in IMS the HP application program can control the IBM. Under TSO and CMS the IBM controls you. It is important when programming with IMF

intrinsics that the programmer know the use of these intrinsics inside and out (the HP IMF class would be helpful). Secondly, the programmer needs to realize that much trial and error will be required in tracking down when and how IBM will respond. But once the series of transmits and receives are worked out the program will work almost flawlessly. Upon completion of the Inventory Control System actual subroutines were developed to handle all HP-IBM/IMS communications. A single call to the subroutine TRANSIMS contains all the logic necessary to send and receive messages while handling all possible error conditions. The data to be sent to the IBM for processing is passed in the subroutine call with the subroutine returning the processed data. Future generations of programmers have been spared countless hours of trial and error testing.

V/3000 - VPLUS/3000 - VIEW/3000 OR WHATEVER HP NOW CALLS IT

V/3000 is used in the Inventory Control System for all screen processing. V/3000 intrinsics are called from the preprocessor program to read, write and edit screens. Since many papers have been presented to the IUG on V/3000 in the past further discussion as to the inner-workings of V/3000 is not really necessary.

The key to using V/3000 in the Inventory Control System is in the require- ment for plant-unique processing. Since each plant is different, many of the differences in processing are simply in the inputting of data. For example, Figure 5 represents two different plant "packing" transactions. (Packing represents the process of taking material produced from a silo and putting it into bags, drums, etc.) V/3000 as a screen processor works very well in this area. Each plant therefore has its own V/3000 forms file to hold its unique forms.

THE PREPROCESSOR

The preprocessor is a COBOL program that runs on the HP/3000. The reason it is called a preprocessor is because it accepts plant-unique data and formats it into common transaction (message) formats for processing by the IBM host. Each plant using the Inventory Control System has its own preprocessor just like it has its own V/3000 forms file. This issue of common transaction formats and preprocessing of data is what ties the system together as it relates to plant-unique processing. Taking our packing example from V/3000, Figure 6 shows the relationship between the plant's unique format and the standard transaction format. This use of standard transaction formats allows the plant, through V/3000 and its preprocesor, to create any type of input screens or put any data relationships, table lookups or special plant coding that is needed into the preprocessor. As long as the proper standard transaction is produced, the plant can do about anything it wants in the preprocessor, including the updating of plant-required data bases resident on the HP/3000.

THE PREPROCESSOR - PUTTING IT ALL TOGETHER

Basically this is how the preprocessor works and interrelates with IMF and V/3000:

1. The user selects a transaction, such as packing, from the master menu.
2. The packing screen is displayed on the CRT and the user inputs the necessary data.
3. The screen is then read using V/3000 intrinsics with V/3000 field edits taking place.
4. If all V/3000 edits are passed the transaction is then put into its standard transaction formats and the IMS transaction code is applied.
5. The transaction is then sent to the IBM mainframe using IMF intrinsics.
6. The IBM mainframe processes the transaction, either updates the data base or gets inquiry data depending on what the user is doing, while the preprocessor waits for a response from IBM that all processing is complete.
7. The transaction is then received by the preprocessor using IMF intrinsics.
8. The preprocessor unformats the standard transaction returned into the plant's unique format.

9. The screen is then redisplayed showing that the transaction was accepted or that it was rejected with the proper error messages displayed in the V/3000 window.

This logic applies to all transactions where one message is sent to the IBM and only one is received. In the case of inquiries one message is sent, such as a request for all open orders, and many messages are received. In this case, during step 7 above, the returned messages are written to a KSAM file as they are received from the IBM host and then unformatted as needed in step 8.

FUTURES

As can be seen from the above steps, true distributed processing is taking place. Data formatting and editing is taking place on the HP/3000 with actual data base interaction taking place on the IBM. IMF/3000 has taken what used to be done in batch mode and allowed it to be done in an on-line mode.

But what is equally important as the use of IMF in distributing Monsanto's Inventory Control System and the potential to use the same techniques in building future systems requiring HP-IBM interaction, is the use of the same concepts in a strictly HP environment. For example, a data base system using IMAGE could be built on one HP/3000 with other HP/3000's accessing and process- ing data from that data base much in the same way as the Inventory Control System does. In MIC these concepts are being studied for use in interconnect- ing HP/150 personal computers to HP/3000's. Additionally, IMF is being used to transmit documents generated from HPWORD to IBM/5520 word processors via the IBM mainframe and CMS.

Concepts are the key, however. The use of the IUG forum to present specific applications and concepts for future systems growth is invaluable.

Michael E. Ura is a Principal Systems Analyst in Monsanto's Industrial Chemical Company located in St. Louis, Missouri. His nine years of MIS background include operations, programmming, systems analysis, project management and consultation. Mr. Ura has a B.S. degree in Business Administration from the University of Missouri and a Masters degree in Management from Webster University.
