A DISTRIBUTED COMPUTER SYSTEM INTERCONNECTING HP3000, HP 1000 AND OTHER MINI - COMPUTERS

BJÖRN DREHER, HANS VON DER SCHMITT, RAIMOND SCHOECK

INSTITUT FÜR KERNPHYSIK DER UNIVERSITÄT D-6500 MAINZ

WEST GERMANY

A Distributed Computer System Interconnecting HP3000, HP1000, and other Mini-Computers

Björn Dreher, Hans von der Schmitt, Raymond Schoeck
Institut für Kernphysik der Universität
D-6500 Mainz, West-Germany

1. Introduction and early history

The Institut für Kernphysik der Johannes Gutenberg-Universität at Mainz, West-Germany, is a medium size institute for basic research in the field of Nuclear Physics. We have an 340 MeV linear accelerator for electrons to perform research in the nuclear structure area using electromagnetic interaction. Currently an 175 MeV two stage c.w. race-track microtron for electrons is under construction, which is controlled and operated with the help of two HP1000 computers. The first of the two stages is operational since 1979.

Until 1976 the control of experiments and the data acquisition was performed by a (today still operational) CDC1700 "minicomputer". In addition a substantial part of the data analysis was also done on this system. By that time we noticed that the computing power and the tools for program development of the CDC1700 were no longer adequate for our tasks. Therefore we decided to purchase a new powerful minicomputer system (HP3000) for the data analysis part of the work and to connect to it several smaller systems (HP1000) as front-ends for the real-time applications.

Since at that time the cost for disc memories was higher than today and memory-resident operating systems were still around, we wanted to be able to perform the program development to a large extent on the HP3000, even for the front-ends. Operating systems were to be generated on the HP3000 and then downloaded to the small system.

In addition, to reduce the cost for the front-ends, these should be equipped only with experiment related peripherals, such as CAMAC interfaces, and not necessarily with expensive magnetic tape transports or line printers. A concentration of those devices at the HP3000 would promise a much higher utilization and availability to all front-end computers.

At that time DS3000/DS1000 was not yet available, but HP had available a product called "Programmable Controller", which was an HP2100 (or at that time already an HP21MX) computer connected to the HP3000 via a 16-bit parallel link using Universal Interfaces at both ends. With this product came cross software that enabled the user to generate RTE-C operating systems, to assemble HP1000 Assembler programs, bring it into an absolute form using a Cross Loader (XL2100) and download it to the front-end computer. There existed also an (unsupported) Cross FORTRAN compiler that was compatible with those days' FTN-IV compiler of RTE-III. Our final configuration is shown in figure 1. The two interconnected HP1000 systems are today running RTE-IV operating systems, the third one used to work with RTE-C and we are currently in the process of rewriting our applications to be compatible with RTE-IV. For the CDC1700 we built an interface to let it appear to the HP3000 like an HP1000 computer, so that we could connect it to a third Universal Interface card.

```
       M i c r o t r o n                E x p e r i m e n t s
          |                  |                  |                   |
          |                  |                  |                   |
   +----------+       +----------+       +----------+       +----------+
   |          |       |          |       |          |       |          |
   | HP1000   |=====| HP1000   |       | HP1000   |       | CDC1700  |
   |          |       |          |       |          |       |          |
   +----------+       +----------+       +----------+       +----------+
                            |                  |                  |
                             \                 |                 /
                              \                |                /
                               \               |               /
                                \              |              /
                                 \             |             /
                           +-----------------------------+
                           |                             |
                           |          H P 3 0 0 0        |
                           |                             |
                           +-----------------------------+
```

Fig. 1: Overview of our distributed computer system (the CDC1700 will be replaced by a PE3220 system).

There existed already some communications software for a similar, but in some essential points different, distributed system at the Technical University in Berlin. This had been written jointly by HP Frankfurt and the Technical University of Berlin. It was kindly made available to us and constituted the basis for our current communications system between the front-end computers and the HP3000.

2. The current system

In the process of getting the initial system running it turned out that both communication drivers in the HP3000 (IOREMO) and in the HP1000 (DVR63) were not suited for our problem. Therefore we had to modify both, especially the HP1000 side. Now the communication between the two computers is really interrupt driven on both sides without the need to loop on a status request to see whether the other side is willing to send, to receive, or to do nothing at all.

Today three computers are connected in a star configuration to the HP3000 and a forth one will be added in the near future. The peripherals of the HP3000 are accessible to the small computers through the file system or directly via special communication drivers in the HP1000 in the case of magnetic tape units, line printer, and plotter. Direct program to program communication between a program on the HP1000 and one on the HP3000 is also available.

In the following we give an overview about the possibilities a user on one of the front-end computers has in the current system:

a. Access to the entire file system of the HP3000

   - Read and write sequentially or in direct access

   - Position to records, write filemarks, space filemarks, rewind, etc.

   - Obtain the status of a device

   A supervisory program in the HP3000 (CENTRAL) keeps track of all files opened by programs in the front-end computers. so that only those programs in the front-end computers may access the files who "own" it. When a program closes the connection to the HP3000, all files opened by it are automatically closed. In addition there is the option to open files globally, so that they can be accessed by more than one program simultaneously.

b. Initiate batch jobs

c. Create and activate processes

d. Program to program communication between programs in the front-end computer and programs in the HP3000.

e. Perform MPE commands

f. Generate an RTE-C operating system on the HP3000 and download it into the target machine. Develop application software in FORTRAN-IV or HP1000 Assembler, bind it into the target system and download it dynamically into the target machine.

g. Transparent use of peripherals of the HP3000. Magnetic tape units and the lineprinter are accessible from the front-end processors as if they were connected directly to them, e.g. through FORTRAN READ/WRITE statements or EXEC-calls. The plotter is available through standard (Calcomp-) calls. This concept is easily expanded to other peripherals.

According to the initial demands to the system, communication is normally initiated by one of the front-end computers. Each request consists of a pair of messages of variable length. The first message contains the request type and the necessary data. The second (return) message contains possible error codes and the resulting data. The interface on the HP1000 side is a set of subroutines (SATTL) that allow the programmatic execution of all features mentioned above, or a direct EXEC call to the drivers of the (virtual) peripheral devices attached to the HP3000. An interactive program (KOPPL) allows to exercise all requests to the HP3000 and to transfer files from one machine to the other.

The receiving process in the HP3000 is one program (CENTRAL), which performs all necessary operations to satisfy the individual requests. This process runs as an always present batch job (in the CS queue), one job per link to a front-end computer. Each front-end computer can have up to 5 programs communicating with the HP3000 at the same time, each of which may have up to 10 files simultaneously open. Those numbers are more or less arbitrarily chosen and can be easily increased when the need arises. Requests from different programs can be freely intermixed.

3. Need for additional functions

As one can see from the previous chapter, the current system constitutes a fixed master/slave relationship between two communicating computers, where the front-end processor is the master and the HP3000 is always the slave. This was satisfying for the first few applications, but soon it turned out that a dynamic establishment of the master/slave relationship and a more direct communication between programs/processes in the various machines would make many applications easier to implement. In particular, it should be possible to start an exchange of messages from any computer.

A general process to process communication across the entire distributed system seemed to be the most attractive solution. Of course, the existing higher-level functions of our current system should not be touched.

4. The HP1000 message system

Fortunately a system that fulfilled many of those needs had been implemented in 1978/79. It was developed in our institute for the inter-process communication (IPC) among the various on-line control programs which are distributed within the two HP1000 computers that control the new microtron mentioned in chapter 1.

The communications protocol of the process-to-process layer is based on the exchange of messages via just two operations: SEND and RECEIVE. Messages have the same form whether beeing exchanged locally in one computer or between the two computers. The participants of the communication are addressed by 10-bytes symbolic names. These are mapped to target computer and process by the system.

Messages are transmitted as sequences of fixed-length packets in a store-and-foreward fashion. Each packet is 64 bytes long consisting of a header (essentially sender and receiver addresses) and the data. The packet transmission constitutes the computer-to-computer protocol, which is thus very simple and easy to standardize.

In addition, I/O requests to devices attached to remote computers are transparently handled by using IPC between I/O drivers.

In summary, this message system enabled us to build a modular distributed control system. The modules (programs) are explicitly portable between the two computers by using the symbolic addressing scheme and by the transparent I/O system. Thus programs can be freely redistributed on demand in a growing system, as our microtron control system is.

5. The second generation communications system

Based on these ideas, we are currently in the process of implementing a second generation of our communications system uniformly throughout our distributed system. It will interconnect one HP3000 with three HP1000's and one PE3220 computer.

The new system will also be an IPC system transmitting messages decompositioned into packets. However as compared to the above application, the system must be capable of higher data flows since the number of data bytes per message will be larger on the average. Therefore the packet size will be increased to 128 bytes. On the other side there is less need for a fully symbolic addressing capability and other overhead-generating features of the HP1000 message system. Thus a higher data throughput may be achieved. Some essential features of the new system will be given below.

From an architectural viewpoint, messages between two processes will be transmitted in our system by a store-and-foreward packet switching mechanism.

Splitting messages into a number of smaller packets has important advantages:

a. Long messages do not necessarily monopolize a communication line. They can be interleaved with packets from other (more urgent) messages.

b. By using fixed length packets as the vehicle of message transfer, it is easy to buffer messages prior to the actual transmission and incoming messages before they are collected by their recipient. Buffering space will be taken from a global packet pool common for all ports of a particular computer.

c. Buffering separates nicely the lower communications layer, that controls just the traffic of incoming and outgoing packets, from the next higher layer, that consists essentially of the decomposition of messages into a series of packets (SEND operation), the reconstruction of messages from a series of packets (RECEIVE operation), and the mapping of symbolic addresses.

d. A store-and-foreward capability comes as a by-product from the buffering.

Besides packeting/de-packeting, packet transmission, and buffer management, the system has to perform additional monitor functions on the participating processes.

i. Processes must be suspended and re-activated in the course of SEND and RECEIVE requests.

ii. Time-out conditions may arise in RECEIVE as well as in SEND requests; the former when an expected message is not received in time, the latter when a transmitted message is not consumed by the recipient in time.

From the hardware point of view we will still have point-to-point connections between the various computers in a star configuration (with the exception that two HP1000 systems are connected directly with each other), the HP3000 being the inner node. We will use the existing hardware (16 bit parallel plus some control lines) with Universal Interfaces in the HP-computers on both sides. However, as can be seen from point c above, the kind of hardware is not that important for the designed function of the whole system.

6. Implementation

To reduce programming time and to improve maintainability and documentation of the system as well as portability we decided to write as many as possible routines in a high-level language. For the non-HP3000 systems we decided to use RATFOR [1], which is a FORTRAN dialect that adds structured elements to FORTRAN-IV. The output of the RATFOR preprocessor is standard FORTRAN-IV, which serves for the portability of RATFOR programs. Only few routines on the HP1000 systems are written in HP1000 Assembler.

The same approach will be taken for the PE3220 system, which will replace the CDC1700 computer.

Since it is expected that the HP3000 will have the highest message traffic of all computers and since several routines have to perform their tasks in privileged mode, we decided to write the HP3000 side in SPL, at least the inner kernel with the most time-critical parts of the system. This allows us to make the best use of the HP3000 architecture and its instruction set, thus lowering the time overhead introduced by the packet switching architecture.

7. Conclusion

In summary, even the current (first generation) system and the dedicated HP1000 message system have proved very valuable in many daily applications. The second generation system, that will be available on all our in-house (mini-)computer systems, will have an even higher impact on many current and future applications. After 4 years of experience with an own, custom designed, communications system, we feel that in our case this was the right way to go. Even today, there is no communications sytem commercially available that would fulfill exactly all our needs. Having all the knowledge about the system in house allows us quite easily to add new required function to the system, as they arise. The fact that most modules of the system are written in a high-level portable language makes it easy to put the system on other computer families and integrate them into our distributed system.

[1] B.-W. Kernighan, P.J. Plauger: Software Tools, Addison-Wesley Publishing Co. 1976