A significant contribution to the high throughput achieved by the cluster comes from the simplicity of its higher level software interfaces.

Operating system function, such as file or device I/O, has associated with it a system service process dedicated to performing a class of functions. Each process has associated with it one or more exchanges, at which requests (which are simply messages in memory of a specified format) from other processes (actually pointers to requests) can be queued. Tables within the operating system are used to determine which exchange is needed for requests of a given type. The system service process obtains a request from its assigned exchange, performs the requested function, moves the completion status code and any response data back into the request block, and then responds to the request, returning it to the exchange of the requestor.

In the cluster environment, this mechanism is modified only slightly by the addition of two agent processes—one at the cluster workstation and one at the master. At the cluster workstation, the agent process receives any requests for resources at the master and maintains an internal queue of such requests. When a poll is received from the master workstation (typically, every 100 milliseconds), the request is moved to a buffer and transmitted to the master. At the master workstation, DMA is performed from the line directly into one of a group of buffers known as transmission buffers. When reception is successfully completed, the master agent process performs a series of link-level and higher-level protocol checks, modifies several fields within the request block to ensure that the response is returned to it, and then issues the request at the master workstation. When the response is received, the master workstation agent process updates the protocol control information in the transmission buffer and at its first opportunity transmits the response to the cluster workstation. The cluster workstation agent then moves the response data and status fields back to the original request block and responds to it, returning it to the requestor.

Summary

The cluster architecture, then, provides a highly cost-effective mechanism for allowing a small number of users to share a common set of resources. In conjunction with the CT Net gateways available as options at the master workstation, it provides all the facilities of fully distributed architectures, while reducing the per-user cost burden to a bearable level.

Barry I. Kelman recently left Convergent Technologies to pursue several independent projects with Santa Cruz Computer, a small consulting firm in Aptos, California. His most recent projects at Convergent involved both hardware and software design in several areas of the N Gen product line, most notably the high-speed cluster communications facility. Previously, he was responsible for the development of much of Convergent’s communications software. Prior to joining Convergent, Kelman spent many years developing communications and terminal software for Perkin-Elmer Corporation’s Data Systems Group (formerly Interdata, Inc.). He has also been employed by RCA and Zentec.

Kelman holds a BA degree in chemical physics from Columbia University and is currently enrolled in the MBA program at San Jose State University.

Kelman’s address is 629 Giant Way, San Jose, CA 95127.