5109-82
ABS-ONLY
175099
N94-22547
p. 1

The VLBA Correlator — Real-Time in the Distributed Era

D.C. Wells (Nat. Radio Astronomy Obs.)

The Correlator is the signal processing engine of the Very Long Baseline Array [VLBA]. Radio signals are recorded on special wideband (128 Mb/s) digital recorders at the 10 telescopes, with sampling times controlled by hydrogen maser clocks. The magnetic tapes are shipped to the Array Operations Center in Socorro, New Mexico, where they are played back simultaneously into the Correlator. Real-time software and firmware controls the playback drives to achieve synchronization, compute models of the wavefront delay, control the numerous modules of the Correlator, and record FITS files of the fringe visibilities at the back-end of the Correlator.

In addition to the more than 3000 custom VLSI chips which handle the massive data flow of the signal processing, the Correlator contains a total of more than 100 programmable computers, 8-, 16- and 32-bit CPUs. Code is downloaded into front-end CPUs dependent on operating mode. Low-level code is assembly language, high-level code is C running under a RT OS. We use VxWorks on Motorola MVME147 CPUs. Code development is on a complex of SPARC workstations connected to the RT CPUs by Ethernet.

The overall management of the correlation process is dependent on a database management system. We use Ingres running on a Sparcstation-2. We transfer logging information from the database of the VLBA Monitor and Control System to our database using Ingres/NET. Job scripts are computed and are transferred to the real-time computers using NFS, and correlation job execution logs and status flow back by the route. Operator status and control displays use windows on workstations, interfaced to the real-time processes by network protocols. The extensive network protocol support provided by VxWorks is invaluable.

The VLBA Correlator's dependence on network protocols is an example of the radical transformation of the real-time world over the past five years. Real-time is becoming more like conventional computing. Paradoxically, "conventional" computing is also adopting practices from the real-time world: semaphores, shared memory, light-weight threads, concurrency. This appears to be a convergence of thinking.