Multi-computer seismic programs

Rick Ottolini

ABSTRACT

Programs can be simultaneously executed on multiple computers to take advantage of the strengths of each computer (e.g. number crunching, graphics). Two schemes on our SEP network are (1) graphics workstations serving a large central computer, and (2) a computation engine serving graphics workstations. The former scheme looks more promising.

INTRODUCTION

We are in the midst of putting a high level graphics workstation (Sun) on each SEP researcher's desk. These are connected to our mini-super computer (Convex) via a standard network (Ethernet). We wish to do interactive graphics and interactive processing of reflection seismic datasets. Since the super-computer is much faster numerically processing data than the workstation and the workstation superior in speed and software tools for interactive graphics, this suggests a multi-computer approach to interactive data processing.

MULTI-COMPUTER PROGRAMS

Generic model

Figure 1 illustrates a generic multi-computer program. A program consists of four parts: (1) setup, (2) dataset management, (3) computation, and (4) interactive graphics. These parts may be distributed among one or more computers in what is called the client-server model. The client initiates requests for services from a service program. Services required by interactive seismic programs include data management, computation, and graphics.

The communications protocol connects parts of a multi-computer program together. The protocol (1) establishes a network connection, (2) transmits and converts data types between possibly heterogeneous computers and (3) has a command language for data, computation, and graphics requests. The communications protocol is typically the least developed piece of software in current computer networks.

Graphics server

One multi-computer program model uses workstations for graphics services. The communications protocol is a network graphics systems. We currently use MIT'S X-Windows on the Convex, Microvax, and Sun 3-160. (Similar systems are expected from Sun and IBM soon.) X-Windows makes the network connection, transmits and executes graphics requests between any source machine and destination display. X-Windows also defines a machine-independent command subroutine library.

Compute server

An alternative model uses a number cruncher for computation services (Figure 3). The *remote procedure call* establishes the network connection and transmits requests and data. Sun Microsystem's Network File System [NFS] implements both remote procedure calls and trans-network data translations. Currently applications programs must define the command language and the process the commands.

Comparison between graphics and compute server models

The compute server command language is more complex than the graphics server command language because it covers more territory. Compute server systems (e.g. NFS) are presently less developed than graphics server systems (e.g. X-Windows), particularly at the command language level. A compute server executes more than just graphics in a workstation, thereby making fuller use of a workstation's capacity and capabilities.

EXPERIMENTAL RESULTS

I have written an application that interactively selects a sub-region of an image and immediately displays the 2-D power spectrum under both multi-program models. I found it easier to write it for a graphics server because (1) it is simpler and (2) our graphics server software is more complete than the compute server software.

Figure 1: Multi-Computer Programs

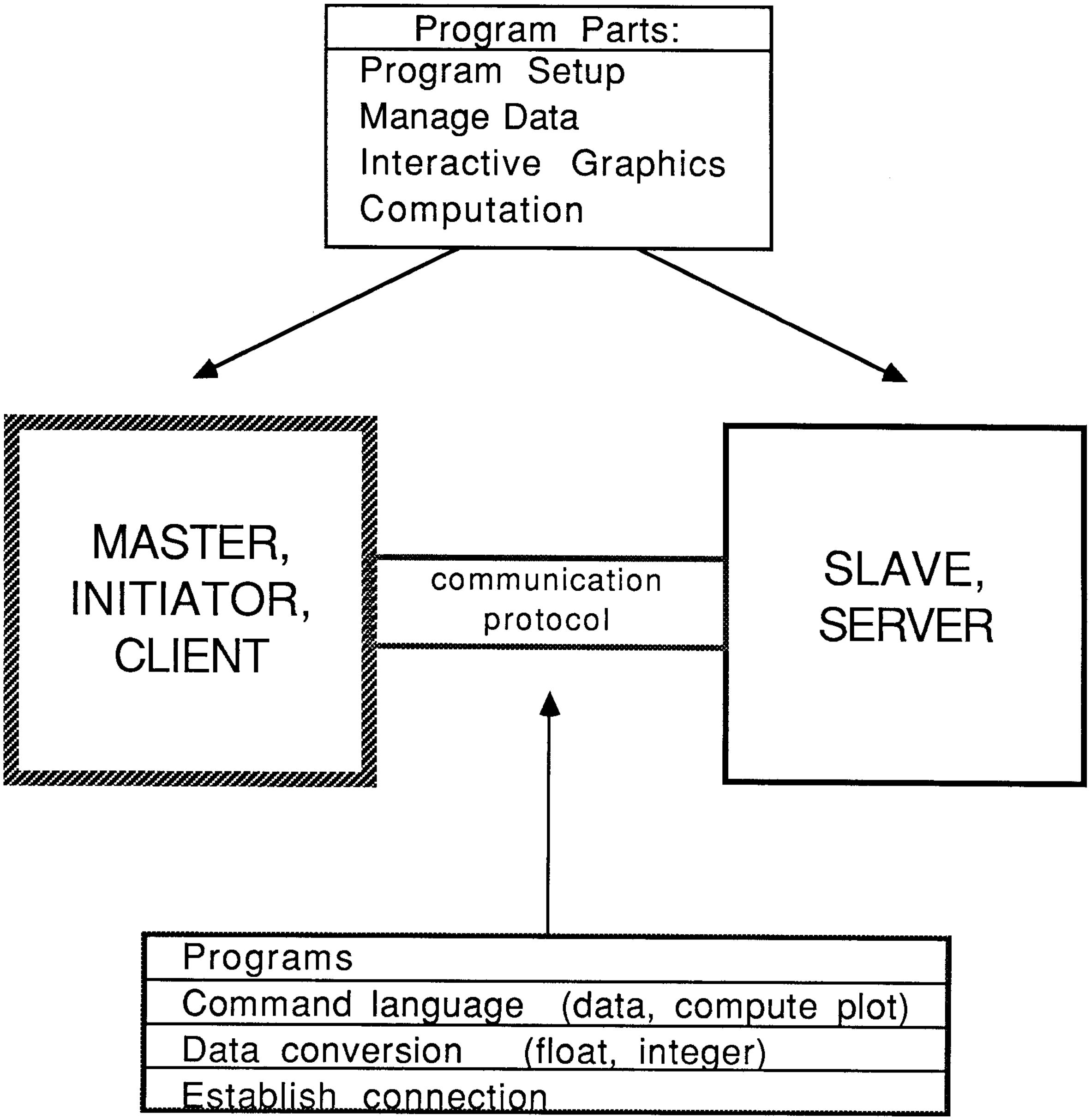
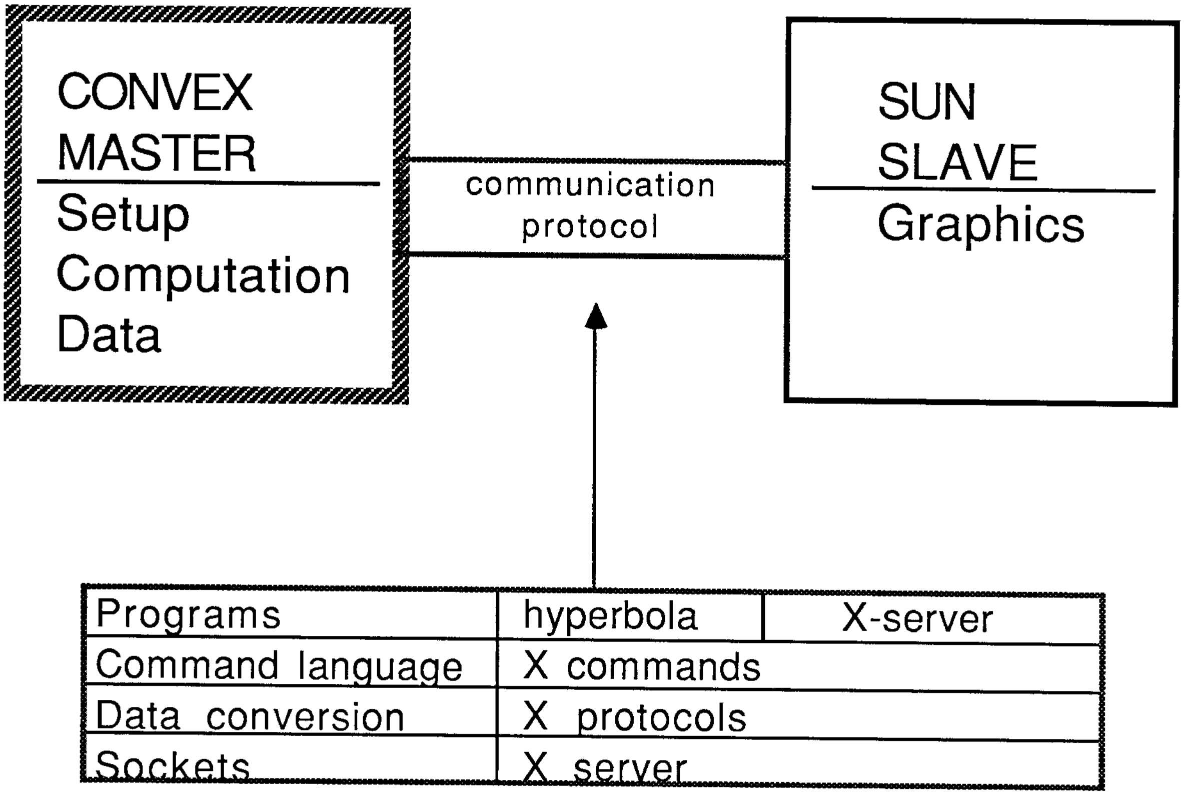
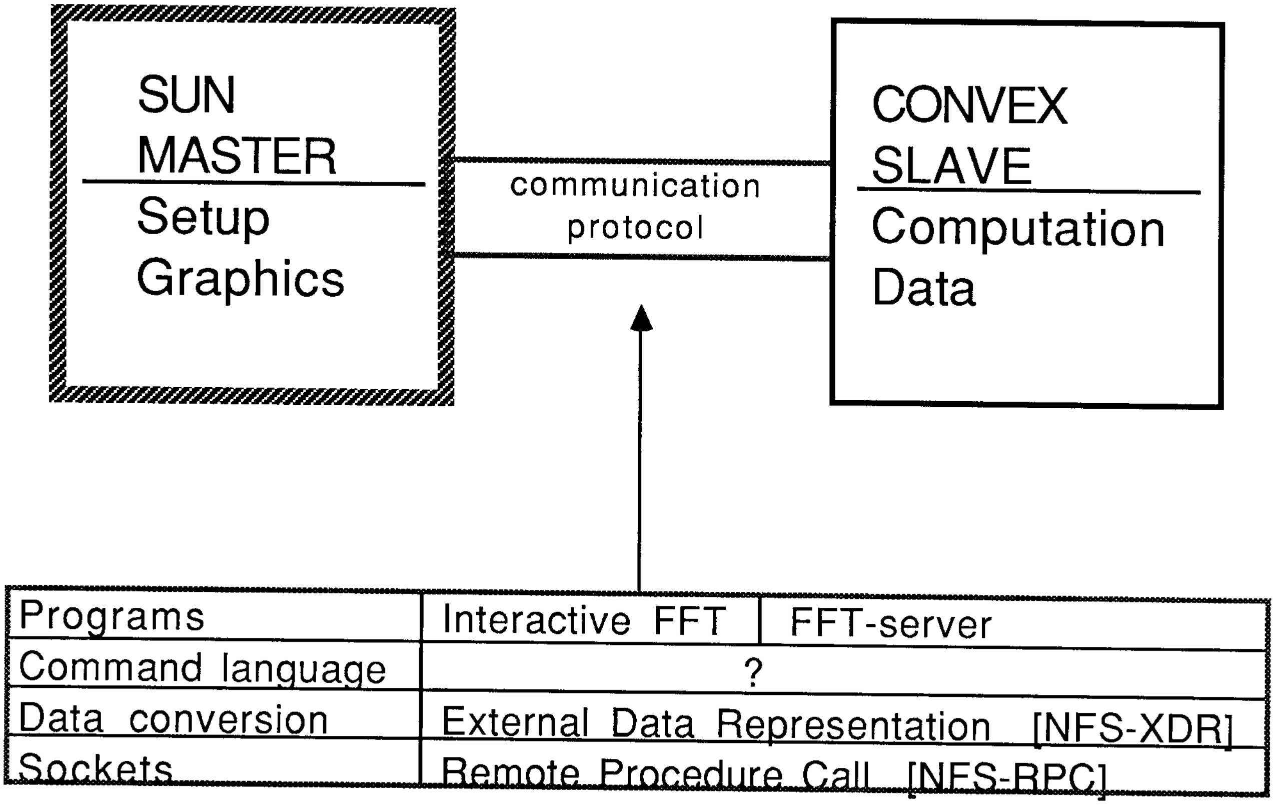


Figure 2: Sun Graphics Server



Programs	hyperbola	X-server
Command language	X commands	
Data conversion	X protocols	
Sockets	X server	

Figure 3: Convex Compute Server



Programs	Interactive FFT	FFT-server
Command language	?	
Data conversion	External Data Representation [NFS-XDR]	
Sockets	Remote Procedure Call [NFS-RPC]