A DISTRIBUTED WEB-BASED METACOMPUTING ENVIRONMENT

Giovanni Aloisio, Massimo Cafaro
Facoltà di Ingegneria, Università degli Studi di Lecce
Lecce, Italy
email: (aloisio,cafaro)@ingle01.unile.it

Roy Williams, Paul Messina
Center for Advanced Computing Research,
California Institute of Technology
Pasadena, California, USA
email: (roy, messina)@cacr.caltech.edu

ABSTRACT

Remote-sensing data about the Earth's environment is being created and stored at an ever-increasing rate. To disseminate these valuable data, they must be delivered in a usable form to those who can interpret them and to those who would learn to interpret them. The aim of the paper is to demonstrate how remote-sensing data can be delivered using a web-browser as a front end, with distributed high performance computing services and replicated data archives. In particular, we show the implementation of supervised on-the-fly SAR processing: this goal can be achieved by an heterogeneous distributed computing environment in which high performance technologies and web interfaces are effectively utilized to provide animated, supervised, custom processed data. The architecture is designed to support both high and low-speed networking, both supercomputers and workstations, the result supporting both professional and casual users.

1. Introduction

The term "heterogeneous computing" has in the past concentrated on connecting different types of computer architectures into a single system capable of useful work. We would like to propose that this term should be extended to cover heterogeneous software packages, connected by heterogeneous communication protocols, and to systems available at heterogeneous bandwidths, from telephone modem to parallel gigabit links.

We would like the system to be useful to heterogeneous users—meaning both different levels of expertise and different levels of access permission, from surfer to professional scientist, archaeologist, etc. We are trying to make SARA (Synthetic Aperture Radar Atlas) [1] a prototype of this kind of system, providing data and computing services with the emphasis on heterogeneity, flexibility, and reliability. The software and protocols that SARA uses include:

- Commercial web servers and browsers running CGI and Java, communicating with HTTP,
- Nexus/Globus: a distributed computing and resource-management protocol [2],
- Postgres: a database management system, communicating through SQL queries and Postgres protocol [3],
- MPI: a conventional message-passing protocol for parallel computing [4], and
- HPSS: a distributed hierarchical storage system manager communicating by DCE [5].

2. SARA: Synthetic Aperture Radar Atlas

The objective of the SARA project [1] is to develop an architecture for a high-speed, high-volume, multi-protocol, distributed database, together with a means to attach distributed computing resources for data conversion, visualization, or knowledge discovery. In pilot form, SARA is a database of 30 Gbyte of multichannel SAR images from the SIR-C Shuttle mission in 1994 [6], (representing less than 1% of the total SIR-C data), distributed among storage systems in Italy (University of Lecce, Italy) and in California (Caltech and San Diego Supercomputer Center), including IBM SP2 systems at Caltech and SDSC, and HPSS hierarchical storage systems at Caltech and SDSC. We are also planning to exploit the Italian facilities available at the Italian Space Agency Center in Matera, the Center for High Performance Applications (CASPUR, in Rome) and the Signal & Images Processing Center at Bari.

SARA provides not only a benchmark for reliability, bandwidth, and security, but also a testbed for protocols and software. SARA has its own specialized web-browser software in Java for viewing and for initiating supervised processing on images from the archives. Output from SARA can be delivered to different devices: a web browser, a cutting-edge virtual-reality device such as a CAVE, or may be post-processed for fusion with other data or computing systems. The connectivity may be achieved by means of a very low-tech channel such as a telephone and modem: SARA is freely available from the Internet (for certain services) [1], with the flow of visitors proving the reliability and security of the infrastructure and its ability to handle multiple requests. SARA is not only a way to deliver Synthetic Aperture Radar images, but also a template for application developers and researchers who wish to create a distributed metacomputer/metadatabase.

2.1 Heterogeneous Protocols

In the past, all computer users stored data in files and directories. Personal computing and the Internet have changed this. The next generation of researchers is comfortable with data access via Web-servers, Web-enabled databases, Intranets, or database protocols such as ODBC or Corba. Programming is with message-passing or High-Performance Fortran, and data is stored in a hierarchical storage manager. We must be ready to send and receive data using these protocols, and we must be ready to address if and how these protocols can be used with communication channels that are high-bandwidth, possibly parallel. If these high-speed communication channels are achieved by an Earth-orbiting satellite, the time to start or stop data flow (latency) may be very high, corresponding to many megabytes “in the pipe”.

While we think of data-intensive computing as a high-bandwidth network linking archives and supercomputers, we recognize that applications will also need access to the low-speed Internet. It may be that the application fuses two data sources, one much more bandwidth-intensive and/or compute-intensive than the other, so that while the large dataset must be ported to high-technology resources, the small dataset may continue to be accessed through the Internet from a remote server. While this may be merely a step on the porting journey, it may also be that creating a web-server on a low-technology machine in the researcher's own office is the most effective way to make this data available to the distributed supercomputer. Thus we would like to be sure that development of high-technology computing resources shares a clear coevolution path with "ordinary" data access methods such as the Web.

2.2 Web Access

The major reasons for using the Web as a transfer protocol are ubiquity, portability, reliability and trust. It is a comfortable, low-tech delivery system that is available at the lab or at home, or from a laptop in a hotel room. While the web may not be suitable for high-speed computing, it is perfect for a quick look, for browsing, or for testing a hunch; this may then generate a request for supercomputer time for a closer look. The web also provides an easy transition for researchers and their existing, highly evolved applications.

The introduction of MIMD computing promised cheap supercomputing for all, but the barrier of human-intensive software porting proved to be a large one, deterring many of those who would have benefited. We must avoid this trap in this next phase of data-intensive computing, by providing a step-by-step path by which researchers can take advantage of high-technology data-handling and computing resources. Rather than demanding simultaneous parallelizing of all code and all data before using the supercomputer infrastructure, we might first provide data-archiving services with Internet access, followed by running the processing software on the supercomputer, followed by parallelizing the processing, followed by parallelizing the data-flow between archive and processor. This step-by-step process of application porting and development requires that any protocol used within the high-speed infrastructure also be available to the external Internet, and that the browser plugins, Java applets, API's and protocol extensions should be implemented in a portable fashion and freely available.

2.3 Servers

As shown in the Figure, SARA distinguishes server processes - boxes with solid outline - from computers, which have the diagonal-striped thick lines. A server provides a transient service, answering a single question with a single answer without changing state, whereas a computer has a persistent state, with persistent communication channels connected to other computers. We have defined three kinds of server:

- Data-servers: the machines closest to the data that serve large binary objects such as images in response to simple requests, with a restricted filtering ability, such as subsetting and compression services.
- Metadata-servers: which respond to complex queries about the available datasets, using a language such as SQL. The server can also provide a list of archives in which the data can be found, with security information and approximate access times.
- User-servers: web-servers that interacts with a user in a sophisticated way to formulate requests to the metadata and user servers. These machines can also serve documentation on the data, and also provide services such as taking in user contributions and maintaining a bulletin board.

These servers, together with the persistent distributed metacomputer (below), are the four elements of the SARA architecture. We feel that the machine closest to the data should be doing minimal computation, except perhaps for compression if the output is low-bandwidth, and consequently the data-server should do nothing but serve data, allowing greatest access to the data itself, with compute services unloaded on other machines. To assure that the data is available to a community of users, the server process(es) that are closest to the data should be carefully checked for maximum reliability: no memory leaks, zombie processes, unclosed files, core dumps, debugging runs, or other unwanted side-effects of immature software.

This is easiest to achieve if

- Users may not make persistent connections to the data server, and
- there are only a small number of server processes, each using a well-known protocol, so that many users can avail themselves of each server.

A similar argument holds for the metadata and user servers if these are public resources rather than private, development machines.

2.4 Computers

In addition to the servers, computers can process and transform the data for visualization and knowledge discovery. For images, these tasks might be familiar image-processing tasks such as interpolation, rotation, contrast manipulation, etc., or more compute-intensive, interactive tasks such as Principle Component Analysis, Singular Value Decomposition, or Maximum Likelihood Classification or fusion with other datasets, such as a digital elevation model.

For the purpose of this paper, we shall concentrate on the processing of raw SAR data. In this case, a large amount of processing is required to convert the signal recorded from the instrument to images; and the majority of the US and European SAR data has not been so converted. A primary objective of SARA is to use the compute services for an on-demand SAR processor, so that a user can order processing of a dataset through the Internet, then come back after an hour or so when it is completed, and retrieve the resulting multichannel images. The computing service may itself be a distributed computer. The computers and data-servers can communicate by a specialized distributed-computing protocol such as Nexus (see below), or by a message-passing library such as MPI; or it may be that the web protocol (HTTP) is the only common language. Output may be fused with data from another source, used as input to a specialized high-bandwidth graphics device such as a virtual environment, or back to the data archive for further, later processing.

Nexus is a protocol that uses remote-service requests to communicate between the machines of the metacomputer. One machine causes a function to be executed on another machine, with a buffer of typed data provided as input—the arguments to the function. The advantage of this approach over conventional message passing is the possibility of multithreading: the machine that executes the function may create a new thread for it while continuing with what it was doing before. In this way we can efficiently utilize both shared and distributed memory machines in the same distributed metacomputer. On the other hand, it may be that conventional message-passing software is already available for some of the tasks of the metacomputer, tasks that are suitable for a homogeneous parallel machine rather than a distributed heterogeneous machine. For this case, Nexus provides a thin software layer that allows MPI and other message-passing libraries to be used without expensive software changes.

The Globus system[8], a companion to Nexus, allows a user to be less specific about which computers do what: the user says what nature of compute resources are wanted, and Globus will find appropriate resources that the user may acquire. We hope to extend this idea to the data resources.

3. SARA Architecture

The three human icons at the left of the Figure represent three ways to use the system. At the top, the simplest way to interact with SARA, a user manages a Web browser to interact with the user server, which in response fetches metadata and data; this browser can of course have helper applications and plugins to allow processing and archiving of results.

In the middle, a specialized, persistent SARA browser is shown, which may be a graphical or text-based interface, and it may be written in C and compiled locally, or it may be a Java applet downloaded from a Web browser. This mechanism provides a way for a mature application such as a GIS or image-processing package to use the data provided by SARA.

At the lower left is an example of an architecture for a distributed, persistent, computing engine connected to the database. The client, by using Nexus, can acquire suitable collections of processing nodes for compute-intensive tasks such as on-demand processing of raw SAR data, which is the purpose of the network in the Figure.

The Synthetic Aperture Radar (SAR) is an active sensor widely used in remote sensing to obtain high resolution ground images from the back-scattered echo signals. When processing is performed, a high computational load is involved: SAR processing is a complex process but it can be simplified considering four main stages: Range compression, Corner Turning, Azimuth compression and Kernel estimation.

The main computational load occurs in the Range and Azimuth compressions, for which circular convolutions in the Fourier domain are exploited, and in the data matrix transposition (Corner Turning) between the compressions. Furthermore, to obtain a good image quality, additional processing has to be performed (Kernel estimation) between the corner turning and the azimuth compressions.

Range compression, Corner Turning and Azimuth compression are generally done sequentially for a given dataset, though parallel/pipelining solutions can provide extra parallelism [9-11]. The range and azimuth compression phases can be easily parallelized by assigning lines of pixels to the processors in a standard domain decomposition. Kernel estimation is a complex process (but essentially serial) requiring specific algorithms, an expert system, a neural net, or even human intervention to be effective.

We indicate parallel dataflows from the data server to the nodes responsible for Range compression, then Nexus remote service requests to communicate the result to the Corner Turning and Kernel Estimation processes, which similarly passes the result to the nodes responsible for Azimuth compression, and thence to an output device or back to the archive. Since range compression and azimuth compression are being run on a homogeneous parallel machine, and the software is already written with message-passing, we have shown these groups of nodes communicating with the MPI protocol.

Conclusion

SARA is a prototype architecture with three major thrusts: data storage, computing services, and access mechanisms. With this pilot system, we hope to demonstrate that all three can be reliably provided for the people that need them [7].

To a large extent, SARA uses Web protocols for data transfer, because the Web has proved itself in terms of ubiquity, portability, reliability and trust. It is a comfortable, low-tech delivery system that is available at the lab or at home, or even from a laptop in a hotel room; furthermore the Web has shown itself able to deal with great heterogeneity, both in terms of the documents and software it delivers, and in terms of the expectations and abilities of its users.

We would like to push these attributes into the high-speed, high-volume computing arena, where possible, while demanding that cutting-edge distributed and parallel software becomes trusted, reliable, even ordinary, for a majority of scientists.

References

[1] R. D. Williams, G. Aloisio, M. Cafaro, G. Kremenek, P. Messina, J. Patton, M. Wan, "SARA: The Synthetic Aperture Radar Atlas": http://www.cacr.caltech.edu/sara/

[2] I. Foster and C. Kesselman, "The Nexus Multithreaded Runtime System": http://www.mcs.anl.gov/nexus/

[3] "Postgres Database Management System": http://www.ki.net/postgres95/

[4] "MPI: the Message Passing Interface": http://www.mcs.anl.gov/mpi/

[5] "HPSS: High Performance Storage System": http://www.sdsc.edu/hpss/

[6] "The Imaging Radar Homepage": http://southport.jpl.nasa.gov/

[7] R. Williams, "Dreaming of Hypermaps": http://www.cacr.caltech.edu/sara/dreaming.html

[8] I. Foster and C. Kesselman, "The Nexus Multithreaded Runtime System": http://www.mcs.anl.gov/nexus/

[9] G. Aloisio, M. Bochicchio, G.C. Fox, R. Albrizio, A. Mazzone, N. Veneziani, "The Design of a Parallel/Pipeline Multiprocessor System for a Fast DFT Algorithms Computation" *Parallel Computing, Problems Methods and Applications*, Ed. P.Messina A.Murli, Elsevier, Amsterdam 1992.

[10] G. Aloisio, R. Albrizio, A. Mazzone, N. Veneziani, "Multiprocessor Architectures for SAR Data Processing" in *Remote Sensing, Global Monitoring for Earth Management*, vol.1, pp.267-270, IEEE 91CH29710.

[11] G. Aloisio, M. Bochicchio, "The use of PVM with workstations clusters for distributed SAR data processing*", Proc. HPCN Europe 1995, Milan, Italy, Lecture Notes In Computer Science, Springer-Verlag, N.919, 570-581, 1995.*

Acknowledgments

The work has been supported by the Italian Space Agency grant ASI-94-RS-169 and by the Center for Advanced Computing at Caltech.

SARA
Architecture

http://www.cacr.caltech.edu/sara