U.S. DEPARTMENT OF THE INTERIOR
U.S. GEOLOGICAL SURVEY

A Parallel-Processing Approach to Computing for the Geographic Sciences

by

**Michael Crane,[1] Dan Steinwand,[2] Tim Beckmann,[2] Greg Krpan,[2]
Jim Haga,[3] Brian Maddox,[4] and Mark Feller[5]**

Open-File Report 01-244

2001

[1] U.S. Geological Survey, EROS Data Center, Sioux Falls, SD 57198-0001
[2] Raytheon ITSS, EROS Data Center, Sioux Falls, SD 57198-0001
[2] Raytheon ITSS, EROS Data Center, Sioux Falls, SD 57198-0001
[2] Raytheon ITSS, EROS Data Center, Sioux Falls, SD 57198-0001
[3] Raytheon ITSS, EROS Data Center/Alaska Field Office, Anchorage, AK 99508-4664
[4] U.S. Geological Survey, Mid-Continent Mapping Center, Rolla, MO 65401
[5] U.S. Geological Survey, Rocky Mountain Mapping Center, Denver, CO 80225

CONTENTS

ILLUSTRATIONS

KEY WORDS

Parallel Processing, Beowulf Clusters, High-Performance Computing, Modeling

ABSTRACT

The overarching goal of this project is to build a spatially distributed infrastructure for information science research by forming a team of information science researchers and providing them with similar hardware and software tools to perform collaborative research. Four geographically distributed Centers of the U.S. Geological Survey (USGS) are developing their own clusters of low-cost personal computers into parallel computing environments that provide a cost-effective way for the USGS to increase participation in the high-performance computing community. Referred to as Beowulf clusters, these hybrid systems provide the robust computing power required for conducting research into various areas, such as advanced computer architecture, algorithms to meet the processing needs for real-time image and data processing, the creation of custom datasets from seamless source data, rapid turn-around of products for emergency response, and support for computationally intense spatial and temporal modeling.

INTRODUCTION

The project described in this paper is a continuation of work that commenced in fiscal year (FY) 2000 with the identification of individuals at U.S. Geological Survey (USGS) Mapping Centers interested in building an information science research infrastructure within the National Mapping Discipline (NMD). At that time, employees at the EROS Data Center (EDC) in Sioux Falls, SD, the EDC/Alaska Field Office (AFO) in Anchorage, AK, the Mid-Continent Mapping Center (MCMC) in Rolla, MO, and the Rocky Mountain Mapping Center (RMMC) in Denver, CO, prepared and submitted a research proposal to begin investigations into high-performance computing. Because the proposal was accepted with a reduced level of funding late in the fiscal year, accomplishments were limited to the acquisition and installation of basic Beowulf clusters. An initial list was compiled of potential applications for implementation on these systems. Red Hat Linux 6.2 was selected as the operating system of choice for the first prototype parallel processing systems. EDC, AFO, and RMMC have chosen to study the Message Passing Interface (MPI) means of internode communication, whereas MCMC is using the Parallel Virtual Machine (PVM) method. A follow-on proposal for continued funding was approved for FY 2001. This has enabled the Centers to make performance and communication enhancements on the existing clusters and to test a variety of applications on these systems.

By focusing on the clustering of low-cost commodity computers into larger (but still low-cost) parallel computing systems, this project will provide a cost-effective way for the USGS to increase participation in the high-performance computing community. These systems provide a robust platform base for conducting research into areas such as advanced computer architecture, algorithms to meet the processing needs for real-time image and data processing, the creation of custom data sets from seamless source data, rapid turn-around of products for the Emergency Response program, and support for computationally intense computer models.

BACKGROUND

In recent years, there has been significant interest in the clustering of low-cost computers into affordable multiprocessor parallel computing systems. These low-cost parallel computers are typically constructed with personal computers either running open-source UNIX-like operating systems, such as the Beowulf concept (Sterling et al., 1998), or using the platform's native operating system, as in Project AppleSeed (Decyk et al., 1999). The most common implementation of these systems is the Beowulf concept, which is typically constructed with commercial off-the-shelf (COTS) Intel Pentium or Digital Equipment Corporation Alpha-based computers running the open-source Linux operating system or

similar UNIX systems.[6] Although the concept of building these systems is no longer cutting-edge research (their construction is well documented – see references), the implementation of applications on these systems is still very much in the forefront of research. Dealing with the increasingly large amounts of data needed to investigate complex geospatial-temporal problems and deciding how to implement these applications in the message-passing environment (Snir et al., 1998; Gropp et al., 1999a) common to Beowulf systems are not well understood.

Network congestion is usually not an issue in parallel processing, as most applications designed to run in this manner are processor bound. In a cluster environment, the only real data being passed are inputs to the various processing nodes. Many parallel-processing algorithms assume that network overhead is a time constant that is so small it has no effect on overall run time. As a result, most of the work in computer science associated with designing algorithms for parallel processing has focused on reducing the run time per processing node. Currently, little attention is being given to data-bound problems related to very large file sizes, such as the geospatial data commonly used by USGS scientists. Most current work only offers advantages for large numbers of small input/output (I/O) requests, such as those found in distributed database systems.

Parallel-processing time, compared with that of a single threaded system, takes the common form of $T = \frac{T_1}{P_{eff}} + c$, where T is the parallel-processing time for a given task, T_1 is the single threaded time for the task, P_{eff} is some coefficient of parallelism, and c is an overhead constant (*Salmon et al., 2001*). In the case of parallel-processing, c is the constant that reflects the time overhead from transmitting data over a network. P_{eff} is a coefficient that is based on the number of nodes in a cluster and the processing power that the nodes have to offer. The effective increase in speed that can be obtained from parallelizing an algorithm can then be calculated as $speedup = \frac{T_1}{T} = \frac{P_{eff}}{1 + \frac{P_{eff} c}{T_1}}$ (*Salmon et al., 2001*). For processor-bound tasks, the amount of time that it takes to transmit data over a network is minimal compared with the amount of time that the processor spends working with those data. In such cases, c can be omitted from the speedup equation, implying that the speedup is mainly a function of the parallelism coefficient.

With data-bound problems, the overhead constant c can take on very large values. For example, 2 GB of information can take anywhere from 3 to 15

[6] Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

minutes to be transferred over a 100-mBit Ethernet network. This is an especially large amount of time when compared with the time that it takes a processor to perform calculations, such as those used in geospatial coordinate system conversions. Because of this, the denominator ($\frac{}{1+\frac{P_{eff}c}{T_1}}$) of the speedup equation takes on very large values and causes the speedup to approach zero. The network overhead then becomes the determining factor in the speedup that a parallel-processing cluster can achieve for data-bound problems. This is similar to what *Dedkov and Eadline (1995)* have observed with data-bound problems running on multiprocessor machines. Their studies have shown that in a multiprocessor system, fast processors actually process large amounts of data less efficiently than slower processors can. This is because the faster processors can overload the system bus with information that can be synchronization information or data passed between processors as part of some algorithm. This communication reduces the bus bandwidth that is available to move data from permanent storage to the processor nodes and back again. Faster processors, then, spend more time waiting on the transfer of data, being in what is known as a "data starved" state. Slower processors have an advantage because they don't pass as much information over the system bus in a given time period as fast processors do. As a result, more bandwidth is available for data to be passed through the system bus.

The implications of having near-supercomputer performance in a package with a price comparable to that of a UNIX-based workstation may be significant to those attempting to use compute-intensive modeling, visualization techniques, and data-rich analysis in their research. These systems will provide cost-effective alternatives to purchasing expensive computing-servers or buying time at supercomputer centers.

Over the course of FY 2001, this project will allow staff or partners to achieve the following:

<u>Develop an understanding</u> of computer clusters and the types of geospatial problems that are best solved on these systems. This includes investigating system hardware, network topologies, system-level software, and applications software. Much of the effort will focus on the programming methods necessary to apply such a system and will also study configuration of the system in relation to the application at hand.

<u>Install a prototype</u> computer cluster at each participating site and develop several examples of applications. These applications will consist of the real-time data processing required for emergency response and scientific visualization; they will run computer programs that model urban growth, land surface trends and processes, and carbon sequestration.

<u>Establish a baseline cluster specification</u> by the end of the project. Identical

systems and a common set of applications and modeling software build toward the concept of an NMD computing "grid" – a networked set of clusters available to deal with our most complex and demanding geospatial-temporal problems.

Create an infrastructure and environment for future research in the information sciences. By fostering a cadre of intellectual talent throughout the discipline, it will be possible within the USGS to investigate higher level topics in information science.

HYPOTHESES

A goal of this project is to evaluate objectively the suitability of Beowulf clusters for supporting the work of the NMD. This will be accomplished by testing several hypotheses about Beowulf clusters. In no particular order, the hypotheses that will be addressed are as follows:

- Run times of numerically intense modeling processes will be significantly faster on Beowulf clusters than on workstation environments currently available to scientists.

- While network and computer configurations are held constant, run times of numerically intense modeling processes can be improved on Beowulf clusters through data-partitioning methods.

- Algorithm-partitioning methods can improve run times of numerically intense modeling applications on Beowulf clusters while holding network and computer configurations constant.

Finding more efficient methods for running models and processing extremely large datasets can aid these programs by offering faster run times and by allowing more data to be processed than is currently feasible on single-node computer systems. This is especially true in cases involving the handling of gigabyte-sized, multitemporal databases, such as those composed of satellite images.

PARTICIPATING SITES

Four centers of the NMD are participating in this project: the EDC in Sioux Falls, SD, which serves as the Nation's archive for land remote sensing data and manages the Landsat 7 satellite system in partnership with NASA; the EDC/Alaska Field Office located in Anchorage, AK, which conducts research and applications projects with a broad spectrum of other Alaska-based Federal and State agencies and international collaborators; and the Mid-Continent Mapping Center in Rolla, MO and the Rocky Mountain Mapping Center in Denver, CO, both of which engage in the generation of cartographic products, research related to cartographic and spatio/temporal data production and application, and

integrated science investigations.

EROS DATA CENTER

The Beowulf cluster at the EDC is installed in the Showcase Visualization Lab, a hub facility that provides centerwide access to high-end demonstration and visualization systems and technical support (fig. 1). Before the cluster was installed, the facilities were enhanced with three additional 120-v, 30-amp electrical circuits and additional cooling capacity. The cluster uses three APS 1400 series uninterruptible power supplies. In addition, shelving was assembled from surplus modular office furniture to hold the main part of the cluster. After initial consideration, it was determined that a noise-reducing enclosure was not needed.

In addition to the Beowulf cluster, the Showcase Visualization Lab contains Windows NT, Macintosh, Sun, and SGI workstations. These computers reside on a subnet that can be isolated from other networks at EDC to facilitate experiments like those proposed in this research.

Figure 1. The 12-node Beowulf cluster at EDC.

To support the clustering project at EDC, a cluster implementation team was formed. This team consisted of researchers from the science group, system programmers, system administrators, and networking experts. As a result of the implementation team's work, the Linux operating system was added to the list of

EDC “supported systems” and in-house expertise was developed. The inclusion of system administrators and networking personnel was necessary to maintain the correct level of system security on the EDC network.

SYSTEM DESCRIPTION

Hardware

EDC’s Beowulf cluster comprises 1 master node and 11 computing nodes, all connected on a dedicated network to ensure the best communications for any given computing task. The master node is a dual processor Dell Precision 420 Workstation running a single Pentium III processor at 733 MHz with 128 MB of RAM bus memory and 50 GB of disk capacity. The computer has two network interface cards (NIC); one is connected to the EDC network, and the other is connected to the dedicated cluster network. The system will be upgraded with a second 733 MHz processor and memory will be brought up to 1 GB. The master node is equipped with a 19-inch color monitor, a keyboard, and a mouse.

The computing nodes consist of 11 Dell OptiPlex GX110 computers, each running a Pentium III processor at 733 MHz with 128 MB of memory and 50 GB of disk capacity. Each computing node is currently equipped with a single NIC and is connected to the cluster network. The computing nodes, which are not accessible to the EDC network, will each be upgraded to 256 MB of memory after a series of benchmarking tests are completed.

The cluster network currently consists of a single 3Comm OfficeConnect 10/100-mBit 16-port switch. In addition, a KVM switch is used, allowing the computing nodes to share a keyboard, video monitor, and mouse for debugging and maintenance purposes.

During summer 2001, the EDC Beowulf cluster will be enhanced with the addition of a data node. The data node is a dual processor Pentium III running at 933 MHz with 512 MB of memory and 144 GB of fast SCSI disk capacity. The system is equipped with a 19-inch color monitor, keyboard, mouse, and dual NICs. Power protection is provided through an additional APS UPS 700. Dedicated data lines will be provided to each cluster computing node and the master node, with the addition of a second NIC in each machine (the third for the master node) and a separate 3Comm OfficeConnect 10/100-mBit 16-port switch. The data node will be connected to the EDC network and will have other data ingest/backup capabilities.

Software

The EDC Beowulf cluster was built using a customized Red Hat 6.2 distribution and Linux 2.2.14 kernal. Both the master node and the computing nodes are running the same level of the operating system - the primary difference is the number of packages installed. The master node is basically a complete installation of Red Hat 6.2, including all compilers and development utilities. The computing nodes have been stripped of nonessential packages and all graphical interfaces.

The system performs its processing using the Local Area Multicomputer protocol, LAM/MPI, version 6.3.1-4. To eliminate duplication of software, we installed the LAM/MPI software on the master node and then shared it with all computing nodes in the cluster through the network file system (NFS).

In order for users to find a consistent work environment, home directories are also stored on the master node and shared through the NFS with the computing nodes. This ensures that the exact same user environment is in place regardless of the node of the cluster a user is on.

During summer 2001, an upgrade from Red Hat 6.2 to SuSe Linux 7.1 will be performed. The SuSe Linux 7.1 distribution will be installed using the Linux 2.4 kernel, which, in addition to various networking enhancements, supports 64-bit architectures and accommodates file sizes over 2 GB. The basic architecture of the cluster will remain the same but will become more streamlined and efficient under the new operating system.

Applications

Two applications are running on the EDC Beowulf cluster – the Clarke Urban Growth Model, December 2000 Release, Beta 3, and the ecosystem model CENTURY.

The Clarke Urban Growth Model is a cellular automata model that was obtained from the EPA, where it was modified to run with the MPI on their Cray supercomputer system. This set of software was compiled with LAM/MPI, and it ran on the EDC cluster with very little effort. Further investigation revealed that several minor modifications and bug fixes were required for accurate and efficient execution on the Linux-based cluster. The MPI Beowulf cluster implementation of the Urban Growth Model was a joint effort between personnel at EDC and RMMC. A dataset for Sioux Falls, SD, has been run through the Urban Growth Model with the EDC cluster configured with 4, 8, and 12 nodes. The model and data were also run on a Sun Ultra 60 workstation that has a price similar to the cluster. Initial results yielded run times of 77 hrs. 30 min., 39 hrs. 52 min., 31 hrs. 9 min., and 1,354 hrs. 39 min., respectively.

The Century Model was developed by the Colorado State University Natural Resources Ecology Laboratory as a general model for use in studying the temporal dynamics of plant-soil nutrient cycling for different types of ecosystems. The model is currently being used to determine the amount of carbon sequestered in the soil of Costa Rica. Initial results indicate a typical sized job should run in approximately 1.5 hours on the Beowulf cluster. The same job would run for 13 hours on just the master node of the cluster and would run for approximately 50 hours on a Sun workstation. To learn more of what is involved in writing a parallel application, see Appendix B, which presents a detailed discussion of the process of implementing the Century Model on the Beowulf cluster at EDC.

Both of these models are in their final-test stage before being turned over to their user groups for running on the EDC cluster. For these research scientists, having their computing-bound models executed on the Beowulf cluster is a significant improvement. They can execute model runs now with various model parameters and data granularities and receive the results in hours, not days or months, of run time.

Applications planned for late summer 2001 include an algorithm for the rectification and reprojection of large image-processing datasets; this will test the feasibility of doing image processing with large image datasets in a Beowulf cluster environment. Three experiments are planned, each testing different distributions of the image datasets. The first will work with a single, large image on the cluster's master node and distribute it among the computing nodes during the processing. The second configuration will test the new data node, using separate communication lines for instructions and data distribution to the compute nodes. The third experiment will test data that is resident on the computing nodes (as might be the case with a data set that is frequently processed to different output geometry). In this case, data communication is reduced to results of the processing only.

Additional models considered for implementation in FY 2001 include the Midcontinent Ecological Science Center’s Invasive Species Model used for ecological forecasting, the National Oceanic and Atmospheric Administration's (NOAA) SLOSH Model that is used to predict storm surge heights and the extent of coastal flooding inundation from hurricanes, and the Spatially Referenced Regressions on Watershed attributes (SPARROW) Model used to simulate nonpoint source pollution loads, such as nitrogen, in the Tampa Bay estuary. The SLOSH Model will be run with an improved integrated topography/bathymetry dataset for the Tampa Bay Basin to evaluate the vulnerability of the population and built environment as part of the USGS’s Tampa Bay Pilot Study. Scientists from other agencies working on the Tampa Bay Pilot Study would also like to have the Beowulf project link the Clarke Urban Growth Model with the Tampa Bay Watershed Loading Model to predict the impact of humans on the natural ecology of the region. In addition, during the

summer of 2001, a faculty member from the South Dakota State University will be investigating a "grid" concept involving the EDC Beowulf cluster, a cluster at South Dakota State University, and two clusters at North Dakota State University. The plan is to take a subset of the existing applications and run them over a wide-area network to test the feasibility of this "wide-area clustering." If successful, other Beowulf clusters in this project will be linked to validate this concept.

ALASKA FIELD OFFICE

The Alaska Beowulf cluster is housed in an environmentally conditioned computer room at the USGS Western Mapping Center's Alaska Field Office in Anchorage, Alaska. Currently, the six nodes making up the cluster are organized on a modular, two-shelf stand as shown in figure 2. All of the equipment in this room is protected by an uninterruptible power supply (UPS). This central location provides easy access for USGS and cooperator staff.

SYSTEM DESCRIPTION

Hardware

The Beowulf cluster at the AFO consists of 1 master node and 5 computing nodes connected on a dedicated 100-mBit local area network. The master node is a Dell Precision 420 workstation configured with one Pentium III processor running at 733 MHz, 1,024 MB of Rambus memory, and 50 GB of EIDE hard-disk capacity. This computer has two 100-mBit NICs; one connected to the local-area network and one connected to the dedicated cluster network by a 100-mBit switch. The master node is being upgraded with an additional 733 MHz processor and a 70 GB SCSI disk drive.

The computing nodes are 5 identical Dell Optiplex GX110 desktop computers. Each computing node has a Pentium III 733 MHz processor, 512 MB of memory, and a 50 GB EIDE hard disk. A 100-mBit NIC is connected to the dedicated cluster network. An additional 40-GB EIDE disk drive will be added to each of these nodes in the near future.

All of the nodes are connected to a single 19-inch monitor, mouse, and keyboard with a 16-port OmniPro KVM switch. The dedicated network consists of a 3Com OfficeConnect 10/100-mBit 16-port switch connected to each node.

Software

The software system is identical to the system at the EDC based on Red Hat Linux 6.2. The Red Hat Linux 7.0 is being evaluated as an upgrade path to the cluster operating system.

Figure 2. At six nodes, the AFO Beowulf cluster makes a compact and space-efficient layout.

Applications

The initial test application at the AFO will focus on using parallel processing approaches for multivariate nonhierarchical statistical clustering of remotely sensed spectral data. Current image-processing configurations generally perform well during multivariate clustering of six-banded Landsat images, but performance drops when the number of image bands or size of the dataset increases. With commencement of the USGS Multiresolution Land Characterization (MRLC) 2000 program, multiple data layers involving up to three Landsat thematic mapper (TM) images and ancillary images (for example, elevation, slope, aspect, and ecoregions) can be processed to derive land cover classifications. Under standard image-processing conditions, up to 20 sample arrays are extracted from a Landsat TM-based dataset for use in an ISODATA algorithm, with the statistical output used to classify the entire dataset. A six-band Landsat TM scene contains about 378 MB of data. If multitemporal Landsat data are used together with ancillary non-Landsat data, the potential

size of the input dataset for clustering could be as many as 23 image bands occupying as much as 1,300 MB of storage space.

The initial test of this application will be to evaluate the speed and efficiency with which the Beowulf cluster can perform multivariate, nonhierarchical clustering on a series of Alaska area datasets. Another test of this application will be to evaluate image classification and data analysis techniques using the Beowulf cluster in the classification and delineation of wetland and upland cover types within a 3.2-million acre study area encompassed by land in the Kodiak Island Archipelago and to perform an assessment of the classification accuracy by analyzing ground reference data. Data for this test will come from multitemporal MRLC 2000, Landsat 7 ETM+ imagery, and ancillary datasets such as slope, aspect, and elevation. Other data, such as Normalized Difference Vegetation Index (NDVI), illumination, and radiation, may be used if initial analysis finds the information useful.

MCMC

The MCMC Beowulf cluster is located in the Geographic Cartographic Research and Application (GCRAS) section's science laboratory in Rolla, MO (fig. 3). This laboratory is a partitioned section of the main Technical Support Services area and is thus partially environmentally controlled. Entry is through a password-protected door in the GCRAS section. This laboratory also houses machines that are used by other personnel for research and applications purposes.

Figure 3. At 17 nodes, the MCMC system is the largest of the Beowulf clusters.

SYSTEM DESCRIPTION

Hardware

The MCMC Beowulf cluster is composed of 17 computer nodes. Sixteen of these nodes were originally surplus equipment that was upgraded to more modern components. Each of the upgraded nodes contains an AMD K6-2 500 MHz processor in an FIC VA-503+ motherboard. They all have 128 MB of system memory. Each has a 9-GB IBM SCSI U2W drive connected to a Tekram

U2W SCSI controller. Also, they each contain some of the original equipment, such as 3Com 3c905 and 905B 10/100 Ethernet cards, assorted video cards, floppy drives, and power supplies. Some of the computers are also equipped with CD-ROM drives that came from the original systems.

The master node is a dual Intel P3-733 MHZ Dell Precision Workstation system equipped with 256 MB of system memory. This computer has two 18-GB U2W SCSI drives connected to an Adaptec U2W SCSI controller card. It has an nVidia Geforce-based graphics card that provides high-end two-dimensional and three-dimensional graphics capabilities. It also uses the drivers from nVidia for hardware-accelerated OpenGL applications. The master node is connected to the network by means of a 3Com 3c905B 10/100 Ethernet card. It also has a 40x IDE CD-ROM drive.

The network is connected by means of a Linksys autosensing Ethernet switch. The cluster network is completely isolated from the MCMC network. This allows research and testing into projects sensitive to network traffic because the internal MCMC network typically carries huge volumes of traffic during the day. It also allows increased security, as it is isolated from the outside.

Software

Each of the refurbished nodes is running a slightly modified version of the Red Hat Linux 6.2 distribution. The modifications included upgrading the kernel to version 2.2.17 and adding some of the vendor-supplied hardware drivers. The install image also made use of updated packages provided by Red Hat. The master node is running the Red Hat Linux 7.0 distribution with kernel version 2.2.18 installed. All of the nodes in the MCMC cluster will be upgraded to the Red Hat Linux 7.x distribution and the 2.4 series Linux kernel. The kernel upgrade is important, as it is hoped that it will resolve some issues that were found in FY 2000 with the way Linux handled virtual memory. Under heavy loads, the kernel's virtual memory system was observed to take unusually large amounts of processor time. All nodes have the standard PVM packages for cluster communication. Also, they all have various GNU development applications installed. The refurbished nodes were all cloned from the same installation and, therefore, are configured in exactly the same way.

The MCMC Beowulf cluster was constructed as part of a Center base-funded project during FY 2000. The goal of that project was to construct a cluster and learn various distributed processing techniques. An application would then be ported to run on the cluster to illustrate the benefits that such a system could provide for the USGS science programs. Construction was completed, and the MCMC cluster went online in March 2000. An MCMC version of the Clarke Urban Growth Model was then selected as the test application to illustrate the benefits of distributed processing. The resulting application allowed the calibrations and prediction of urban growth in Chester County, PA, to be

performed in less than 3 working days. The single-threaded version of the code took 43 days of continual processing to finish just the first calibration phase out of the three necessary before prediction could be done.

Initial work is in progress to convert the MCMC FY 2000 experimental projection code to process data by scanline chunks. This code originally dispatched work on an individual scanline basis, and computing nodes would then process and return their scanlines to a master node for writing to the output file. The problem with this was that the nodes were processing scanlines in the same general area, causing all of the nodes to keep reading the same areas of the input file to produce their output scanlines. This resulted in a lot of duplicated I/O requests. The scanline method also resulted in a large amount of network collisions as the nodes were trying to return their data to the master node at the same time. A chunk-based system will allow the nodes to process larger parts of the file, resulting in more of the file being processed simultaneously than is the case with normal scanline processing. Varying the chunk size will also be checked to determine the effects on network congestion by varying the times when the computing nodes finish their scanline chunks.

A second phase will implement a system based on a processing partitioning scheme. In this scheme, the data are served from a Parallel Virtual File System (PVFS) drive housed across *x* data nodes. A PVFS drive can be thought of as a network-based RAID system. The rest of the computing nodes in the cluster are grouped into *x* processing partitions, with one of the data nodes assigned to each partition. The input file will be spread evenly across the PVFS drive. Each processing partition will be responsible for processing $\frac{1}{x}$ parts of the file and will get most of its data from the PVFS node assigned to that partition. Within each partition, the individual nodes will then work to perform the processing. This scheme is presented in figure 4 below.

The advantage in this setup is that, in the best and the average case processing, each partition should only access the PVFS node assigned to it. Although there will still be network collisions within each partition, at least one node in each partition should be able to perform either a read or a write operation on the partition's assigned file part. This means that at any given time, multiple parts of the entire input file are being read and multiple parts of the output file are being created. This should be an advantage over simple file serving from a single-server node.

Additional research will be done with the Flat Network Neighborhood (FNN) architecture that has been researched at the University of Kentucky. This scheme uses multiple Ethernet cards in each machine and multiple Ethernet

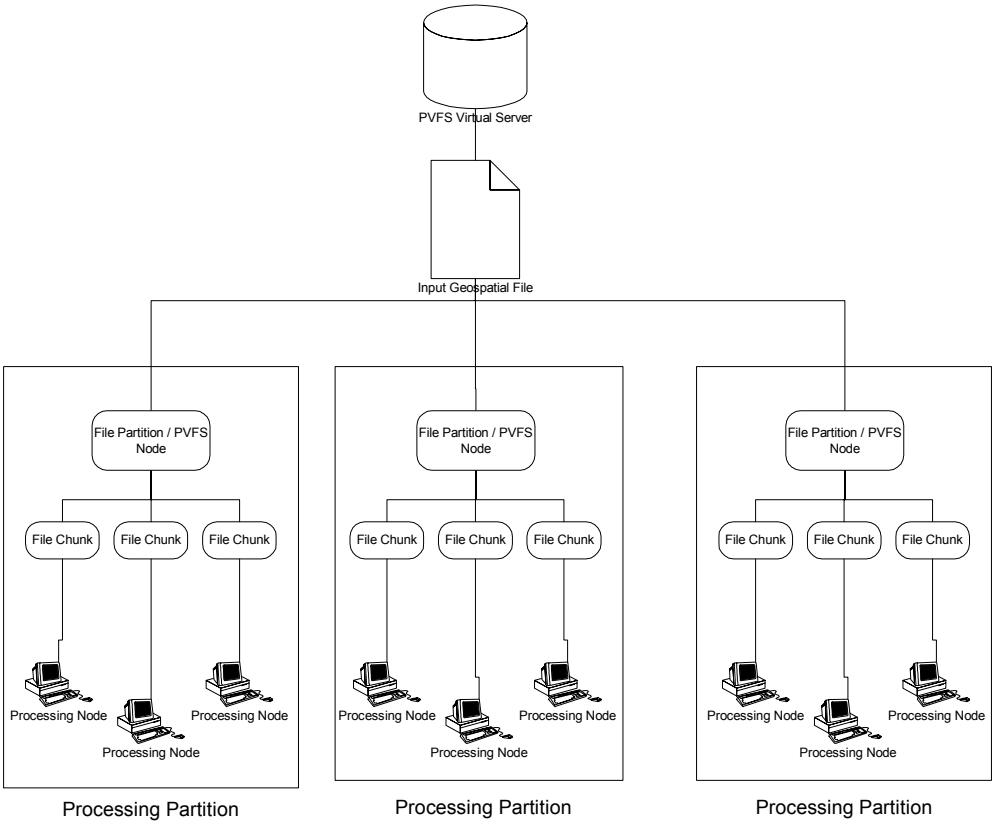

Figure 4. The PVFS Processing Partition Scheme.

switches to connect the nodes. In a normal setup, large numbers of machines are set up on a network with multiple Ethernet switches. These switches are connected to each other through uplink ports on each switch. The disadvantage with this setup is that only one communication channel between switches can be active at any given time. This can introduce a significant bottleneck to processing large datasets, especially if the computing nodes are on a different switch than the data nodes. With the FNN setup, each computer on the network only has to go through one switch to communicate with any other node. This setup may also provide other benefits, such as more overall bandwidth than can be provided by a single Ethernet switch.

For all test phases, initial mathematical descriptions are being developed and will be refined as the project progresses. These descriptions will allow the methods to be compared on a mathematical basis and should allow determinations of the best, worst, and average cases for each method.

Applications

The MCMC part of the project will involve continued research into a distributed geospatial data projection project that was MCMC-funded in FY 2000. This project involves examining methods to project very large amounts of data quickly. The method examined included distributing mathematical interpolations of coordinate system conversion equations to find the best combination of speed and accuracy.

MODFLOW, a USGS Water Resources Discipline model used to characterize the quantity and quality of ground water, is being obtained to run on the cluster. This application requires a distributed system to execute, so it should be well suited to the MCMC Beowulf cluster.

The Clarke Urban Growth Model is used for predicting urban expansion at regional scales. The model can be run independently of other models or used as a module that links other environmental factor models to produce data for investigating the impact of urbanization on the environment. The model was constructed using a cellular automaton that simulates an urban growth process. Cellular automata are used for simulating complex systems using simple models. The types of growth rules defined for the model include spontaneous neighborhood growth, diffusive growth and spread of a new growth center, organic growth, and road-influenced growth. There is a second level of growth rules, termed “self-modification,” that influence the results of the model. These rules are prompted by unusually high or low growth rates whose thresholds are defined by the modeler. Input requirements include temporal road and urban areas, average percentage of slope data, defined excluded areas, and a background or hill-shaded image. There is an option for land use data, and the Anderson Level I Classification has been used to date. All data input to the model must be in the GIF image format. The model execution is done in different phases, with the calibration phases being the most CPU-intensive part. Because the model was being rewritten to take advantage of parallel processing, we chose this application to benchmark our system.

RMMC

RMMC has two Beowulf clusters, known as Jekyll and Hyde, and both are situated in Building 810 on the Denver Federal Center in Colorado. The Jekyll Beowulf cluster is installed in a secure computer room that is environmentally conditioned. The Hyde cluster has been set up in the office environment of the Information Technology Resource Laboratory. Both clusters are using 4-tier wire racks to support the 16 nodes that make up each system (fig. 5).

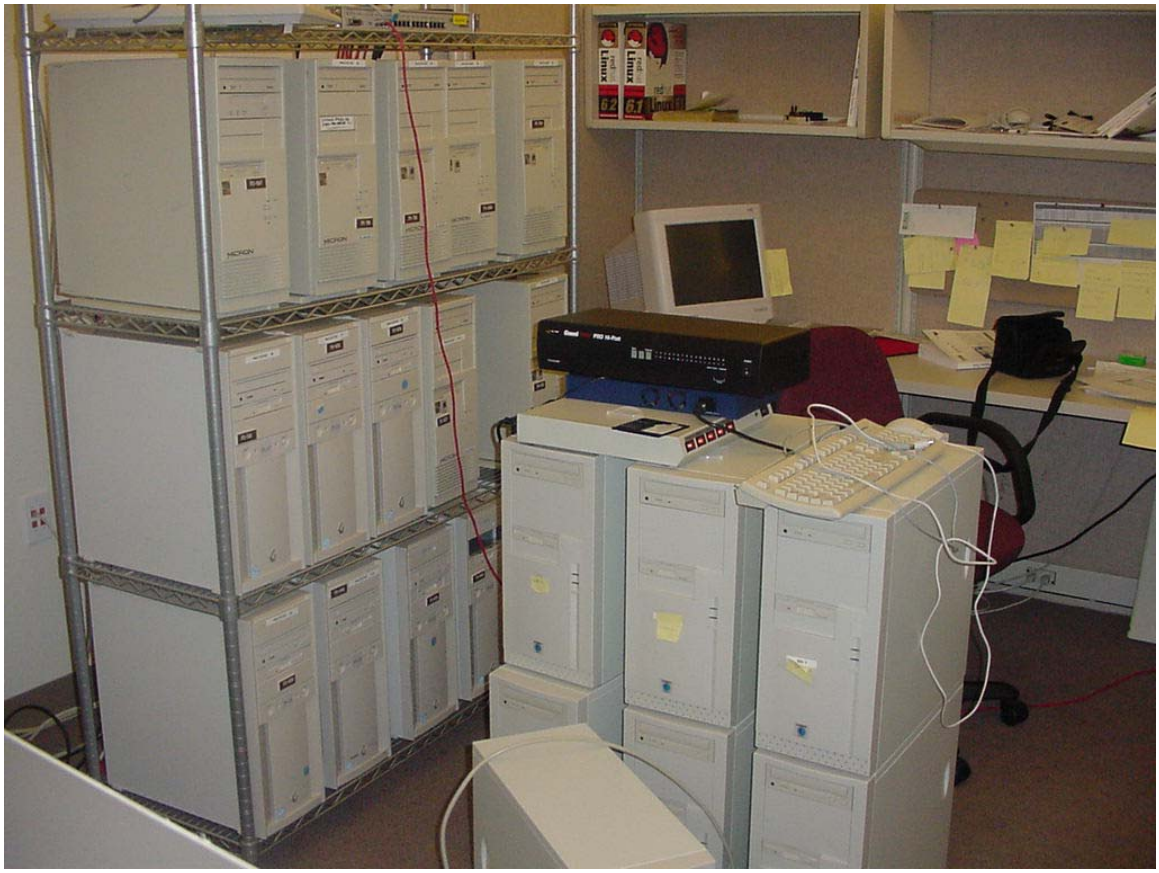

Figure 5. Parts of the Jekyll and Hyde Beowulf clusters before installation in their respective locations at RMMC.

SYSTEM DESCRIPTION

Hardware

RMMC's Jekyll cluster was created from 16 older PCs that were being surplused by the USGS. Jekyll contains 1 master node that consists of a single 200-MHz Intel processor, 6 computing nodes that also have the 200-MHz Intel processor, and 9 computing nodes that contain the 133-MHz Intel processor. The master node has 128 MB of EDO RAM, one 4-GB EIDE hard drive, a 3.5" floppy drive, a CD-ROM drive, two NICs (one is a 3COM 905b, the other a 3COM 509), a 15" monitor, a keyboard, and a three-button mouse. A 3COM, dual-speed, 16-port switch connects one of the network NICs on the master node, and the 15 computing nodes to a dedicated network. The other network card is connected to the RMMC network but is placed outside the firewall.

The 6 computing nodes that contain the 200-MHz Intel processor each have one 3COM 905b NIC, one 2-GB EIDE hard drive, 128 MB of EDO RAM, one 3.5" floppy hard drive, and one CD-ROM. The 9 computing nodes with the

133-MHz Intel processor each have one 3COM509 NIC, one 1.2-GB EIDE hard drive, EDO memory ranging from 64 MB to 128 MB, one 3.5” floppy disk and one CD-ROM. The system was configured so that the 16 NICs connected to the 3COM switch are not accessible to the RMMC network.

The second system, dubbed Hyde, was procured through this project and also consists of 1 master node and 15 computing nodes. Since other systems in the project were using Intel architecture, a conscious effort was made to investigate a different processor type, which is the reason the second system has AMD processor architecture. According to published reports, the AMD processor performed mathematical functions faster than the Intel processor. The master node has an ASUS A7V PC133 Athlon motherboard, with 200-MHz bus speed, an AMD 750-MHz Thunderbird CPU with L2 on-Die cache. There are two 512-MB nonregistered, non-ECC, PC-133 SDRAM DIMMS. The system can be expanded to a total of 1.5 GB of memory. The master node was built with a 30.6-GB IDE, 7,200-rpm Seagate hard drive, one HP9200I 32x/8x/4x rewriteable SCSI CD drive, one Sony 48x IDE CD-ROM, one 3.5” floppy drive, two Kingston EtherX 10/100 NICs, one Viewsonic 21” monitor, an AGP 32-MB Visiontek video card, a keyboard, and a three-button mouse.

The 15 computing nodes have the same motherboard and chip set as the master node. Each computing node has one 512-MB nonregistered, non-ECC, PC-133 SDRAM DIMM, one Sony 48x IDE CD-ROM, one 3.5” floppy drive, one 15.3-GB IDE 7,200-rpm Seagate hard drive, one Kingston EtherX 10/100 NIC, and an ATI Expert 98 8-MB AGP video card.

Software

The Hyde cluster is currently running a customized Red Hat 6.2 Linux distribution on the master node, which came with kernel 2.2.14. RMMC has since upgraded the kernel to 2.2.17. The master node has printer support, an X Windows system, graphics manipulation, a networked workstation, authoring and publishing tools, emacs, a kernel development package, a clustering tool, and standard development and utility packages.

The master node has the TripWire security package installed and is using IPChains as its firewall. The XV was also downloaded and installed for image analysis. MOSIX, which is a software package that enhances the kernel, allowing a cluster of X86 workstations and servers to function cooperatively as a single system, was installed on the cluster. The PVFS has been loaded on the system but has yet to be installed. It is hoped that when PVFS is installed, it will alleviate the I/O bottleneck that can occur on the parallel system.

The cluster has LAM/MPI 6.2.3, MPICH 1.2.0, and PVM 3.4 installed using NFS to ensure consistency to all nodes. The system is using LAM/MPI 6.2.3 to do its

distributed processing, although RMMC is in the process of upgrading the protocol to LAM/MPI 6.5.1.

The computing nodes are running the standard workstation installation of Red Hat 6.2 Linux with the upgrade of the kernel. Directories where the utilities and required programs reside are all obtained through the NFS mount from the master node, guaranteeing that the same packages are executed throughout the cluster.

Enhancements to the system will involve upgrading the operating system on all machines from Red Hat 6.2 to Red Hat 7.1, which includes the kernel 2.4. The major improvement to the cluster should be in the ability of the kernel to scale onto symmetric multiprocessor systems. The kernel subsystems under 2.4, such as networking stack and file I/O, are fully multithreaded. The new kernel is geared for the enterprise. These enhancements should better serve the clustering systems.

APPLICATIONS

The Clarke Urban Growth Model is being run on the more robust Hyde cluster to predict urban expansion associated with the metropolitan areas of Seattle, WA and Albuquerque, NM. The Clarke Model can be run independently of other models or coupled to other environmental factor models to investigate the impact of urbanization on the environment. As mentioned previously, the Clarke Urban Growth Model is based on a cellular automaton to simulate growth. The kinds of growth rules defined for the model include spontaneous neighborhood growth, diffusive growth and spread of a new growth center, organic growth, and transportation-influenced growth. A second level of growth rules, termed self-modification, is prompted by unusually high or low growth rates whose thresholds are defined by the modeler; these rules influence the results of the model. Input requirements include temporal road and urban areas, average percentage of slope data, defined excluded areas, and a background or shaded-relief image. There is an option for land use data, and USGS Anderson Level 1 data have been used to date. All data input to the model must be in a GIF image format. The model execution is done in different phases, with the calibration phases being the most CPU intensive. This application has been chosen to benchmark the Hyde cluster.

RMMC scientists also hope to run the USGS Water Resources Discipline's MODFLOW Model on the Hyde cluster. MODFLOW is a three-dimensional finite-difference ground water flow model whose modular structure enables it to be easily modified to adapt the code for specific applications.

FUTURE DIRECTIONS AND ACTIVITIES

This project is still in its formative stage; hardware and software procurements needed to make the four participating sites fully operational have just been delivered. Nevertheless, the participants believe there are significant rewards to be gained by continuing to explore methods for efficiently processing large amounts of data over a distributed processing system. There are many methods that could be used, and indepth research alone could take over a year before any implementation begins.

Advances in networking technology, such as FNN architectures, promise to provide higher bandwidth, lower latency connections for this type of processing. FNN architectures might also be modified to provide multiple input and output channels per node. This would allow a node to send and receive data over the network more quickly than is possible with a single network adapter.

Other network transmission protocols also need to be studied. The standard protocol, TCP/IP, tends to incur an overhead cost whenever information is sent out over the network. Alternative protocols, such as those studied by members of the PVFS team, may be applicable to this kind of work. These types of protocols allow data to be sent over the network in a different manner than that used by TCP/IP. The benefits from this could be more efficient data transmissions between the nodes.

Gigabit Ethernet systems may provide better methods to send large amounts of data through a network. Currently, however, such systems place too large a load on the computer's processors to be of much use in numerically intensive processing environments.

Another important test that needs to be conducted is wide-area clustering, or a grid concept, between the cluster centers participating in this project. This would involve the incorporation of two, three, and all four sites in a test running the same model with the input data coming from a single site.

Reprojection of large geospatial datasets, detection of change between multitemporal data, specific feature identification and extraction, and other processes associated with digital map revision will be given high priority in FY 2002.

BIBLIOGRAPHY

SELECTED REFERENCES

Anderson, P., 1999, *The Texas Tech Tornado Cluster: A Linux/MPI cluster for parallel programming education and research*, *in* Crossroads, Association for Computing Machinery, Issue 6.1, pp. 28-32.

Brown, R.G., 2000, *Maximizing Beowulf performance*: Durham, NC, Duke University Physics Department.

Brown, R.G., 2000, *Engineering a Beowulf-style computer cluster*: Durham, NC, Duke University Physics Department.

Decyk, V.K., Dauger, D.E., and Kokelaar, P.R., 1999, *How to build an AppleSeed: A parallel Macintosh cluster for numerically intensive computing*: presented at the International Topical Conference on Plasma Physics, Faro, Portugal, September 1999.

Dedkov, Anatholy F. and Eadline, Douglas J., 1995, *Performance considerations for I/O dominant applications on parallel computers*: Paralogic Corporation, ftp://www.plogic.com/pub/papers/exs-pap6.ps.

_____, 2001, *FNN: Flat neighborhood* networks: University of Kentucky, http://aggregate.org/FNN.

Geist, A., Beguelin, A. et al., 1997, *PVM: A user's guide and tutorial for networked parallel computing*.

Gropp, W., Lusk, E., and Skjellm, A., 1999a, *Using MPI: Portable parallel programming with the message-passing interface*: MA, MIT Press, Cambridge.

Gropp, W., Juss-Lederman, S., Lumsdaine, A., Lusk, E., Nitzberg, B., Saphir, W., and Snir, M., 1999b, *MPI: The Complete Reference: Volume 2, The MPI Extensions*: MA, MIT Press, Cambridge, MA.

Hargrove, W.W., and Luxmoore, R.J., 1998, *A new high-resolution national map of vegetation ecoregions produced empirically using multivariate spatial clustering*, URL: http://www.esd.ornl.gov/~hnw/esri98/.

Hargrove, W.W., and Hoffman, F.M, 1999, *Using multivariate clustering to characterize ecoregion borders*: Computing in Science & Engineering, v. 1, no. 4, p. 18-25.

Hargrove, W.W., and Hoffman, F.M., 1999, *Multivariate geographic clustering using a Beowulf-style parallel computer*, URL: http://research.esd.ornl.gov/~forrest/pdpta-1999/.

Hargrove, W.W., and Hoffman, F.M., 2000, *An analytical assessment tool for predicting changes in a species distribution map following changes in environmental conditions*: Fourth International Conference on Integrating GIS and Environmental Modeling, Banff, Alberta, Canada.

Hoffman, F.M., and Hargrove, W.W., 1998, *Making soup from stones*: Troubleshooting Professional, v. 2, issue 5.

Hoffman, F.M., and Hargrove, W.W., 1999, *Parallel computing with Linux*: Crossroads, v. 6, no. 1.

Hoffman, F.M., and Hargrove, W.W., 1999, *Multivariate geographic clustering using a Beowulf-style parallel computer*: Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '99), v. III, CSREA Press, p. 1292-1298.

Hoffman, F.M., and Hargrove, W.W., 1999, *Cluster computing: Linux taken to the extreme*, Linux Magazine, v. 1, no. 1, p. 56-59.

Kittel, T.G.F., Rosenbloom, N.A., Painter, T.H., Schimel, D.S., Fisher, H.H., Grimsdell, A., 1999, *The VEMAP Phase I Database: An integrated input dataset for ecosystem and vegetation modeling for the conterminous United States*, http://www.cgd.ucar.edu/vemap/user_guide.html.

Ligon, W.B., and Ross, R.B., 2000, *An overview of the parallel virtual file system*: Parallel Architecture Research Lab, Clemson, SC, Clemson University.

Mahinthakumar, G.F., Hoffman, F.M., Hargrove, W.W., and Karonis, N.T., 1999, *Multivariate geographic clustering in a metacomputing environment using globus*: Proceedings of the ACM/IEEE SC99 Conference, Portland, OR.

Matthew, N., and Stones, R., 2000, *Professional Linux programming: Databases, PostgreSQL, MySQL, LDAP, security, device drivers, GTK+, GNOME, Glade, GUI, KDE, Qt, Python, PHP, RPC, diskless systems, multimedia, internationalization, CORBA, PAM, RPM, CVS, Flex, Bison, Beowulf, Clustering, ORBit, MPI, PVM, and XML*: Wrox Press.

Salmon, J., Savarese, D. F., and Sterling, T., 2001, *Managing high-volume astronomical data with heterogeneous Beowulf clusters*: Center for Advanced Computing Research, http://www.cacr.caltech.edu/Publications/techpubs/PAPERS/cacr176/cacr176.html.

Snir, M., Otto, S., Huss-Lederman, S., Walker, D., and Dongarra, J., 1998, *MPI: The complete reference: Volume 1, The MPI Core*: Cambridge, MA, MIT Press.

Spector, D.H.M., 2000, *Building LINUX clusters*: Sebastopol, CA, O'Reilly and Associates.

Sterling, T., Cwik, T., Becker, D., Salmon, J., Warren, M., and Nitzberg, B., 1998, *An assessment of Beowulf-class computing for NASA requirements: Initial findings from the First NASA Workshop on Beowulf-class clustered computing*: IEEE 1998 Aerospace Conference, Snowmass, CO.

Sterling, T.L., Salmon, J., Becker, D.J., and Savarese, D.F., 1999, *How to build a Beowulf: A guide to the implementation and application of PC Clusters*: Cambridge, MA, MIT Press.

Thakur, R., Gropp, W., and Lusk, E., 2000, *MPI-OI: A standard, portable API for high performance parallel I/O*: Argonne, IL, Argonne National Laboratory.

Red Hat, Inc., 2000, *The official Red Hat Linux getting started guide: Linux 6.2*: Durham, NC, Red Hat, Inc.

_____, 2001, *The parallel virtual filesystem project*: Clemson, SC, Clemson University, http://parlweb.parl.clemson.edu/pvfs/.

Wang, P., Cwik, T., and Green, R., 2000, *Benchmarks for AVIRIS algorithms on a Beowulf parallel computer system*: AVIRIS Earth Science and Applications Workshop, Pasadena, CA.

Welsh, Dalheimer, and Kaufman, 1999, *Running LINUX*: Sebastopol, CA, O'Reilly and Associates.

IMPORTANT WEB SITES

http://www.beowulf.com – provides a history of the Beowulf project, a partial list of other Beowulf Web sites to check, cluster-related projects, and commercial Beowulf Web sites.

http://www.epcc.ed.ac.uk/epcc-tec/documents.html – the Web site for the Edinburgh Parallel Computing Centre provides a list of free publications and lecture notes and a calendar of upcoming events.

http://clusters.top500.org - an electronic newsletter that contains articles about benchmarks, Beowulf clusters, hardware and software, press releases, and editorials.

http://www.euroben.nl/ - provides benchmark programs for scientific and technical computing to assess the performance of computers for these fields.

http://www.nrel.colostate.edu/projects/century/ - provides a description of the Century Model and its applications.

APPENDIXES

Appendix A: Minutes of the Beowulf Clustering Team Meeting and First Quarterly Report

On January 9, 2001, the U.S. Geological Survey, National Mapping Division's Beowulf Clustering Team held their first meeting at the EROS Data Center. Team members from the EROS Data Center (EDC), Rocky Mountain Mapping Center (RMMC), EDC-Alaska Field Office (AFO), and Mid-Continent Mapping Center (MCMC) attended. Primary topics of discussion focused on each Center's cluster implementation and planned upgrades, impediments, applications to be run, and planning and scheduling cooperative tasks for the remainder of the FY 2001 project.

Attending:
Mark Feller, RMMC, 303-202-4277, mrfeller@usgs.gov
Brian Maddox, MCMC 573-308-3508, bmaddox@usgs.gov
Jim Haga, EDC/AFO 907-786-7035, haga@usgs.gov
Chris Kotyk, EDC/AFO 907-786-7030, ckotyk@usgs.gov
Mike Crane, EDC 605-594-6041, mpcrane@usgs.gov
Dan Steinwand, EDC 605-594-2557, steinwand@usgs.gov
Charlie Trautwein, EDC 605-594- 6015, trautwein@usgs.gov
Tim Beckmann, EDC 605-594-2521, tbeckman@usgs.gov
Mike Neiers, EDC 605-594-6834, neiers@usgs.gov
Greg Krpan, EDC 605-594-6854, krpan@usgs.gov
Brian Davis, EDC 605-594-6856, bdavis@usgs.gov
George Xian, EDC 605-594-2599, xian@usgs.gov
Shuguang Liu, EDC 605-594-6168, sliu@usgs.gov

1. Discussion of each site's hardware configuration – what is running and what is not.

EDC Cluster Overview
- 12-Node Pentium III 733 MHz-based system with 256-MB RAM & 50-GB disk capacity per node
- Have LAM/MPI installed and are successfully using it
- Running Urban Growth Model/MPI beta version from EPA with modifications

RMMC Cluster Overview
- Overview of "built-from-cast-off" system
- Overview of new system procured in FY 2000 (Athalon-based)
- New system waiting for facility mods (power)
- Successful with LAM/MPI
- Running Urban Growth Model/MPI beta version from EPA with mods

MCMC Cluster Overview
- Built from components in FY 2000
- 16-node system, running mainly PVM

AFO Cluster Overview
Similar to EDC but with eight nodes
Not yet fully functional

The need to use common Beowulf benchmarking programs and methods was discussed. Mark Feller will check on appropriate software to use. The four clusters need to be benchmarked before system performance is optimized and before applications are run in order to establish a performance baseline.

Parallelization Methods: Process Parallel, MPI, PVM, Mosix, Gnuqueue, and others?

2. The following applications were identified for the various sites:

1. RMMC and EDC have agreed to standardize on the MPI/EPA version of the Clarke Urban Growth model.
2. MCMC will run a modified version of the Clarke Urban Growth model.
3. A prototype of the Century Model was made in support of a study in Costa Rica, and EDC will use the code base to extend the application for a much larger study area in the United States.
4. EDC and AFO will test data-resident processing-on-the-fly.
5. MCMC and EDC will be performing Geometric Transformations of very large data sets.
6. AFO and RMMC are looking into applications using MODFLOW, a USGS model used to characterize ground water quantity and quality.
7. EDC is investigating the NOAA SLOSH Model that is used to determine the height of storm surge and extent of flood inundation for possible support of USGS activities associated with the Tampa Bay Pilot Study.
8. EDC will collaborate with BRD and run dispersion models for invasive species.
9. RMMC will look into aspects of the map revision process that are suited to parallel processing.

It was agreed that RMMC and EDC would share the Urban Growth Model code and compare general performance on these systems. AFO may also be interested in an application of the Urban Growth Model. MCMC expects to conclude its Urban Growth modeling effort by the third quarter FY 2001.

3. Schedule/Time-Line Development

The project will produce quarterly reports to keep the Centers and Reston

Headquarters apprised of progress, changes in plans, and any problems that may arise. These meeting-minutes will serve as the first quarterly report.
General time schedule:

RMMC & EDC plan to get the UGM 3 beta model loaded and running with minimal modifications to the EPA version during February 2001.

EDC will focus on data resident processing-on-the-fly during February-March.

MCMC will work on geometric transformations of very large datasets now; EDC will collaborate as time permits.

AFO will concentrate on getting its cluster functional and decide on an initial application to run.

Mike Crane needs input for FY 2001 procurements by January 30, 2001.

4. Actions:

   a. EDC is to accommodate RMMC and AFO requests for logins on the EDC cluster.
   b. EDC will look into obtaining surplus machines to build a four-node cluster that will enable Mike Neiers to investigate MOSIX or other Linux kernel modifications on a separate system and, thus, avoid affecting MPI and process-parallel programming research.
   c. EDC will also see if additional machines are available for AFO, RMMC, and MCMC.
   d. EDC will investigate switch speed and determine whether participants should be concerned about this (answer appears to be “no”).

Appendix B: High-Level Design Document for the Costa Rica Stochastic Century Model Implementation on a Cluster of Computers

by Tim Beckmann

1. Scope

This document focuses on the design details of the existing Costa Rica Stochastic Century Model implemented on a Beowulf cluster of computers at the EROS Data Center (EDC). More specifically, it covers those factors that affected the conversion of the previously existing single-processor application into one that could be run in parallel on a cluster of computers to achieve a significant reduction in the time required to run the model. The design of the model is not covered.

1.1 Background

The Costa Rica Stochastic Century Model (referred to as the “Century Model” in this document) was developed and distributed by Colorado State University. The model is used in a program developed by Shuguang Liu, a scientist at EDC, to perform stochastic modeling of carbon sequestration in the soil of Costa Rica (simply referred to as the "program" in this document). A full run of the program executes the Century Model 20 times for each of 1,300 sites in Costa Rica. The original program took 40 hours to run on Dr. Liu’s workstation. The goal of this project was to convert the program to run on the Beowulf cluster and obtain a significant improvement in processing speed.

The Beowulf cluster at EDC consists of 12 personal computers running the Red Hat Linux 6.2 operating system. Currently, each computer contains a single 733 MHz Intel Pentium III processor and 128 MB of Random Access Memory (RAM).

2. Referenced Documents

Costa Rica Stochastic Model paper at:

http://www.Colorado.EDU/research/cires/banff/upload/34/

Century Model Web site at:

http://www.nrel.colostate.edu/projects/century/

LAM/MPI Web site at:

http://www.mpi.nd.edu/lam/

3. Design Consideration

This design document covers only the parallel implementation of the program. It does not describe the serial version of the program or any algorithms of the program except where the parallel implementation is affected.

3.1 Assumptions and Dependencies

It was assumed at the beginning of the project that each of the 1,300 sites was independent of the others and that each site could be run independently. This was confirmed by Dr. Liu before the project began and reconfirmed by the developer through experimentation.

3.2 General Constraints

The Century Model code is written and maintained by students and professors at Colorado State University. The program currently uses version 4 of the Century Model software. Version 5 of the software is available and is a total rewrite of the code. Because it has not been ported to the Unix platform yet, one of the goals of this project was to parallelize the program without changing the Century Model software.

3.3 Tool Availability

The initial phase of the project dealt with research to determine what tools were available for a Linux cluster that would allow programs to be easily parallelized. Several tools were investigated, but in the end, the LAM/MPI implementation was selected. The factors that solidified this decision were the stability and maturity of the LAM MPI implementation, the fact that MPI is a portable standard, and the performance and ease of use of the API.

3.4 Common Source Code

To make the source code easy to maintain and test, the developer intended to keep the serial and parallel code versions maintained in the same set of source code.

4. Design Details

The following sections give details on the design issues that arose and how each was approached. This document discusses only the design issues of the main application.

4.1 Separation of Work

One of the key issues in parallelizing a program is determining how the work that

needs to be completed can be efficiently split up between different CPUs (in this case, nodes in the cluster). To achieve the greatest speed increase possible, choose work packets that are as independent from other work packets as possible. This reduces the communication overhead between the nodes and the time lost waiting at synchronization points between the nodes.

Another factor that has to be considered is the granularity of the work. If each work packet takes a long time to complete on a node, it can adversely affect the overall run time of the algorithm. For an extreme example, assume a given job takes 5 hours to run in a serial implementation. Running an ideal parallel implementation of that job on a four-node system could reduce the run time to 1.25 hours. If the job is split into five work packets that each take an hour to run, the job will take 2 hours to complete. Obviously, the granularity of the job size is too large to make proper use of the system.

For this application, a natural level exists for splitting up the work. Each of the 1,300 data sites is completely independent of the others so each work packet consists of one data site. Each site takes approximately 1.5 minutes to process. While this time is somewhat long, it is significantly less than the 1.25 hours it takes to run the entire job on the cluster and represents a good tradeoff between design complexity and work packet granularity.

4.2 Master and Slave Roles

To efficiently apply the nodes in the cluster to the work at hand, we chose one node as the master node. The master node's responsibility is to split the work into separate packets, assign each packet to a single slave node, and collect and collate the results from the slave nodes. Each slave node is responsible for processing its work packet, returning the results to the master node, and accepting the next work packet.

4.3 Initializing the Cluster

This section describes coding aspects of the design. For details on running the process, refer to the installation documentation. It is important to understand that an application is provided by the LAM/MPI implementation to start a copy of the process on each node in the cluster.

In an MPI application, the cluster is initialized with a pair of MPI API calls. The first is the MPI_Init call. This routine initializes the MPI environment and must be the first MPI routine called by an application. The second MPI call is the MPI_Comm_rank call. This call returns a number indicating which instance of the process this is. A zero indicates that this instance is the master node. Any other number indicates that the instance is running on a slave node.

4.4 Master Node Design

The first thing the master node does is call the MPI_Comm_size routine to determine the number of slave nodes that are available to perform work.

4.4.1 Work Packets Creation

The master node reads the file that describes all 1,300 of the sites (freq.txt). This file is organized so that each line contains the details on a single site. The entire file is read at the start of the process to verify that its contents are valid before work is sent to the slave nodes. This is done to prevent an easily identifiable problem in the file from being discovered after most of a job is complete.

4.4.2 Initial Slave Work Assignments

Initially, the master node sends each of the slave nodes a packet of work. The work is sent using the MPI_Send routine, specifying the particular slave node to which the work will be assigned. The master node keeps track of the site assigned to each slave node.

4.4.3 Collecting Results from Slave Nodes

After the initial assignment of work, the master node waits for a slave node to return results. This is done by calling the MPI_Recv routine to wait for results from any of the slave nodes.

When results are received, the master node determines which work packet they are associated with by checking which work packet the slave was assigned. One thing to keep in mind is that there are no guarantees that the order in which work is sent to slaves will be the same order in which the results are returned. This means that the master node needs to keep the results and write them to the output files in the same order as the serial implementation would have. The merge_buffers routine was created to manage this process. After the results have been sent to the merge_buffers routine, the next work packet is sent to the slave that returned the results and the master loops back to wait for the next set of results to arrive.

4.4.4 Collecting the Final Results

After the master node has sent the last available work packet to a slave node, it keeps collecting results from the slave nodes until all the results have been returned. When all the results have been received, the master node sends a message to all the slave nodes to tell them it is time to terminate. All the nodes call the MPI_Finalize to shut down the MPI environment and then exit.

.

4.4.5 The Merge Thread

To keep the slave nodes busy, the master node must send them more work as quickly as possible. Merging the results from the slave nodes can be a time-consuming process that could delay sending new work to the slave nodes. To minimize the delay, the master node creates an additional thread that is responsible for merging the results. The main thread of the master node is dedicated to communicating with the slave nodes and only needs to perform a very quick operation to let the merge thread know a new set of results is available for writing. The MPI library is not thread safe and all MPI calls must be performed by the same thread.

4.5 Slave Node Design

The slave nodes initialize the MPI environment by calling MPI_Init and MPI_Comm_rank to learn their roles in the cluster, much like the master node. The slave nodes know they are slaves since the return value from MPI_Comm_rank is nonzero for a slave.

A slave node first initializes its working directory by copying some data files from the NFS-mounted disk (see the Directory Structure section). It then falls into a loop where it calls MPI_Recv to receive messages from the master node. The message can either be a packet of work or a message telling the slave that it can exit. If the exit message is received, the slave exits the loop, calls MPI_Finalize, and terminates.

When a work packet is received, it is processed in the same manner that the serial implementation processes a single site. The only difference is that when the slave node finishes processing the site, it sends the results back to the master node by calling MPI_Send instead of writing the results directly to a file.

4.6 Directory Structure

This process requires both a directory on an NFS-mounted drive and a directory that is unique to each node. The NFS-mounted directory is used to store the executable programs and the input dataset. All nodes need access to these files. The actual data transferred from the NFS-mounted directory are just over 1 MB, including the executables and data.

The directory that is unique to each node is used as a temporary storage area. As each site is processed, it writes temporary files to the disk. These temporary files use the same names on each node, and therefore each node needs a unique directory. On our cluster this is easy to do since each node in the cluster has local disks mounted in locations with the same path names. For example, /data2/tbeckman is the local directory I use when running the process. It is unique to each node.

The results at the end of the process can be found in the local directory on the master node, which is the same directory in which the process is started.

5. Acronyms

API - Application Programming Interface
LAM - Local Area Multicomputer
MPI - Message Passing Interface Standard
NFS – Network File System