JPL HOME EARTH SOLAR SYSTEM STARS & GALAXIES TECHNOLOGY

NASA's HPCC
ESS
EARTH & SPACE SCIENCES PROJECT

Introduction
Reports
JPL Activities
Additional Links
Home

Jet Propulsion Laboratory
California Institute of Technology

at the Jet Propulsion Laboratory

ESS Annual Reports for FY2001

This annual report encompasses the JPL activities in the HPCC Earth and Space Sciences Project from October 2000 to September 2001.

Applications

- HPC for NanoElectronics Modeling
- Parallelization of the Regional Ocean Model System (ROMS)
- Distributed In-situ Mission Simulation
- Mars Beowulf
- Convection in Spherical Shells [Thermal_Convection]

Testbeds

- **Beowulf Cluster-class Computing Investigation**
  - Evaluation of Gigabit Networking for Beowulf-Class Cluster Computing
  - Interface Framework (Wiglaf)
- SGI Origin 2000

System Software Research and Development

- PYRAMID Parallel Unstructured AMR Library
- Remote Interactive Visualization and Analysis of Earth and Planetary Image and Terrain Dataset
- Information Power Grid Applications Technology Development Task

Basic Research

- Phase 3 HTMT Architecture and Gilgamesh Transition

*The Goddard Space Flight Center (GSFC) serves as the lead center for the ESS Project and collaborates with the Jet Propulsion Laboratory (JPL).

Page last modified on 12/28/01

HPC for NanoElectronics Modeling

FY2001 Annual Report

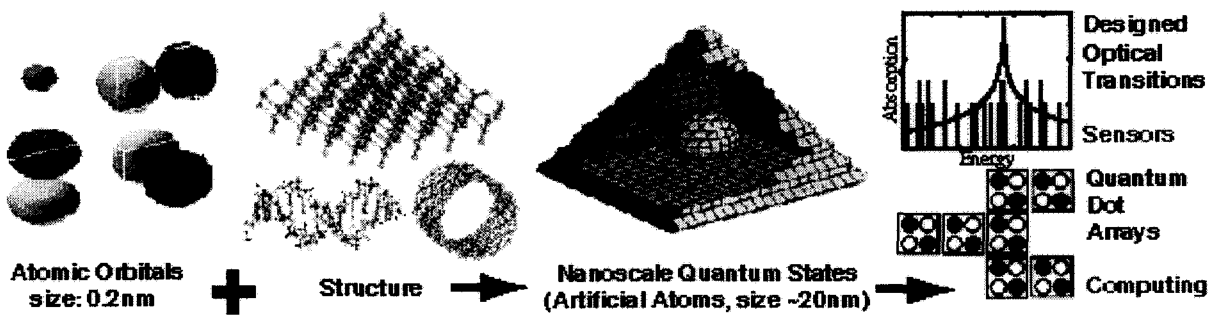


Figure 1

Objective

The objective of this task is to develop a simulation tool for massively parallel platforms that predicts the electronic structure of quantum dots. Quantum dots promise to provide extremely dense, low power computation allowing for highly automated missions and crucial on-board digital processing capability. Quantum dots also offer a viable solid-state technology for 2-5um wavelength detection and emission required to detect life-signature molecules. Eventually, it is expected that the ultimate technologies for computation and detection will be molecular-based. An accurate, physics-based simulation tool provides a means to inexpensively design and test a large array of different bulk material and molecular structures. Such a tool will be essential to insure a successful transition from the current state of the art to a deliverable technology.

Approach

To solve a quantum mechanical problem one must first choose a set of Hamiltonian basis functions which can represent the essential physical properties of the materials comprising the device under investigation. Quantum dots are a demanding application requiring atomic scale spatial discretization and the incorporation of local electrostatic potentials. With these attributes in mind, we choose the empirical tight-binding basis set which consists of s, p and d like orbitals centered on each atom in the structure. This basis set incorporates full crystal symmetry and allows for the investigation of effects that are fundamentally atomistic in nature, e.g. grading, impurities, vacancies, and anti-sites. A key benefit of choosing an atomic orbital basis is that both semiconductor and molecular-based technologies can be investigated with one simulation tool.

As indicated above in **figure 1**, the eigenvalues and eigenvectors of the structure Hamiltonian are used as a basis to calculate observables of interest for hardware applications. At the core of the simulator is an iterative eigensolver capable of handling double complex matrices whose dimensions are on the order of tens of millions.

Accomplishments

This year's efforts were focused on three software development and application issues:

1. We turned our Lanczos-based and Rayleigh-Ritz-based eigenvalue solvers into production code.

   This has enabled us to solve for the electronic structure in multi-million atom systems.

2. Parallelization of the NEMO 1-D code.

   The parallelization of the 1-D transport code enabled us for the first time to study hole transport in realistically sized structures.

3. We continued work on the efficient computation of material properties such as bandgaps, effective masses under strain conditions.

   This enabled the proper material parameterization of binary compounds and ternary alloys using the previously developed genetic algorithm approach. The proper material representation is needed for the modeling of heterostructures such as quantum dots or heterointerfaces.

These three issues are detailed to various degrees below.

Lanczos-based eigenvalue solver parallelization.

In FY00 we reported our first scaling results of our Lanczos eigenvalue solver. The code was tested on an SGI Origin 2000 and a Beowulf cluster. Memory and algorithm issues limited us in performance to about 30 CPUs. The Origin 2000 showed a significant performance reduction if the algorithm was run on more than 35 CPUs. We have eliminated these bottlenecks as shown in **figure 2** below.

Figure 2A displays the performance of 30 iterations of the Lanczos solver on a Beowulf system consisting of 32 nodes with two 933 MHz Pentium III CPUs and 1 GB of RAM per node. Different processors communicate using a non-shared memory MPI implementation and over a slow 100 base-T ethernet connection. Curves corresponding to five different problem sizes are shown. Dashed curves indicating a linear (ideal) scaling are also shown for reference. The two largest problems require enough memory that they require more than one processor to avoid swapping. We chose to implement a nearest neighbor CPU communication system which limits the 0.25, 0.5, 1, and 2 million atom simulations to 32, 40, 51, and 63 CPUs respectively. Efficiency, the ratio of speedup to ideal speedup as a function of the number of processors, decreases with increasing network size. This decrease corresponds to a fraction of unparallelized code of approximately 1.6%. However, **figure 2B** also shows that, for the regime of interest, the efficiency is independent of problem size. This result suggests that while communication bandwidth is not a limiting factor for this problem, there may be some inherent load imbalance in the computation. This issue is under continued investigation.

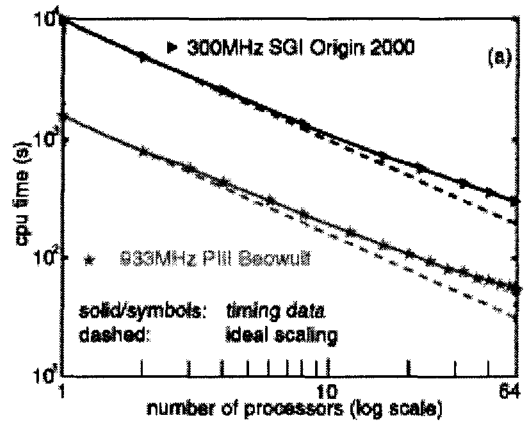


Figure 4A
(click for enlargement)

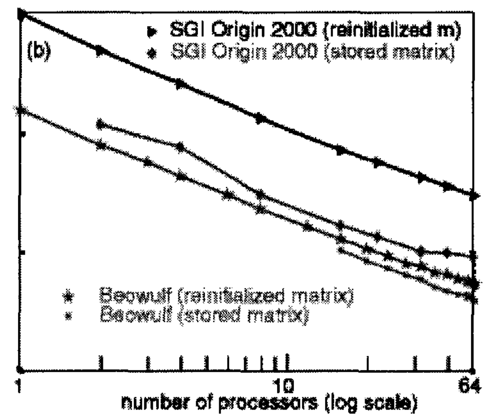
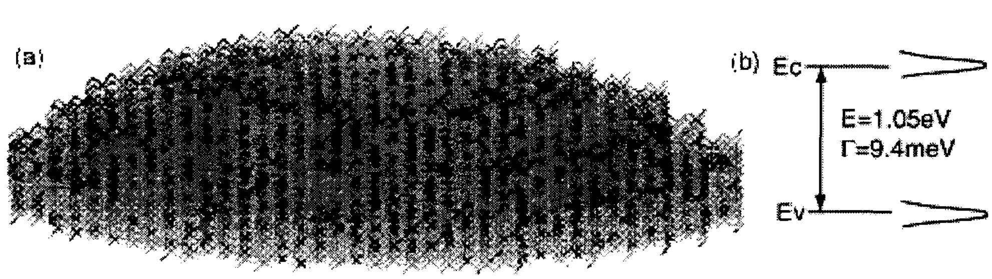


Figure 4B
(click for enlargement)

Simulation of an alloyed quantum dot system

Since we represent each individual atom in the quantum dot system explicitly, we can demonstrate this capability by simulating an alloyed quantum dot system in a GaAs matrix. The dome shaped quantum dot has a diameter of 30nm and a height of 5nm. A 5nm GaAs buffer surrounded the quantum dot in the simulation. Since the In and Ga ions inside the alloyed dot are randomly distributed, different alloy configurations exist and optical transition energies from one dot to the next may vary, even if the size of the dot is assumed to be fixed. We therefore try to answer the question: What is the minimal optical linewidth that can be expected for such an alloyed dot, neglecting any experimental size variations? A side view of such an alloyed quantum dot, which is half the size of the system considered here without the surrounding GaAs, is shown in **figure 5** below. In simulations of 50 random alloy configurations, we have obtained the single particle electron and hole ground state energies and the optical transition energy from their difference. The mechanical strain is minimized using a valence force field method which considers contributions to the total strain energy due to bond length changes, as well as bond angle modification. The mechanical strain field is recomputed for each alloy configuration. For these particular simulations we used the sp3s* basis set, where the matrix elements scale with respect to the equilibrium position with the ideal exponent 2. The simulation of 50 different alloyed dots shows a mean optical transition energy of 1.05eV and a standard deviation, or associated linewidth of 9.4meV compared to experimentally reported linewidth of 34.6meV. The experimental data does of course include quantum dot size variations as well. We plan to simulate larger samples that do include quantum dot size variations in the future. The major result of this simulation is the observation that there will be a significant optical linewidth variation due to alloy disorder alone, even if all the quantum dots were perfectly identical in size.



In0.6Ga0.4As Lense Shaped Dot

Figure 5

Parallelization of the NEMO 1-D code

NEMO 1-D's main task is the computation of current-voltage (I-V) characteristics for resonant tunneling d... ...ıd sp3s* tight bindin... ...ransverse momentum over a transport kernel at each bias point. A full charge self-consistent simulation within this model on a single CPU is prohibitively expensive, as the generation of a single I-V would take about 1-2 weeks to compute. Simpler charge self-consistent models derived from a parabolic transverse subband assumption, eliminating the numerical momentum integral, had been used in the past. Computation on a parallel computer now enables the thorough exploration of quantum mechanical transport including charge self-consistency effects in the whole Brillouin zone. Various parallelization schemes (fine, coarse, and mixed) have been developed and are evaluated in their performance. This effort resulted in two journals and one proceeding publication.

Figure 6A below shows the total time for the computation of an I-V without charge self-consistency (only semi-classical charge self-consistency) as a function of number of CPUs used in the parallel algorithms (run on the 933MHz Pentium III cluster mentioned above). Ideal performance is depicted as a straight line on a log-log scale. 70 bias points (I), 21 k points, adaptive E grid. Parallelization in I, k, and E. **Figure 6B** shows the speed-up due to parallelization compared to the single CPU performance.

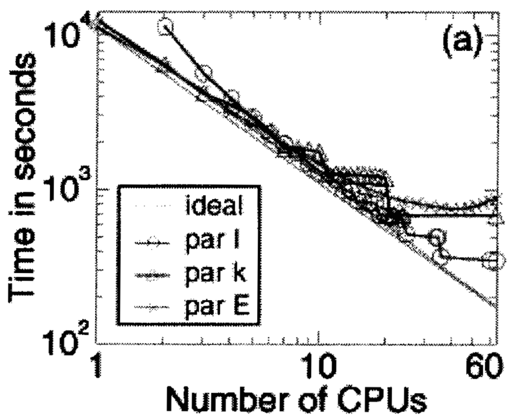


Figure 6A
(click for enlargement)

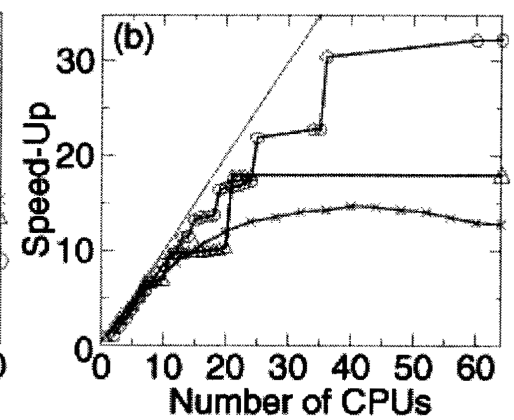


Figure 6B
(click for enlargement)

The coarse and the medium grain (I and k) parallel schemes show significant load balancing problems for larger numbers of CPUs in a realistic I-V computation. The fine grain parallelism (E) is communication limited and load-balancing limited. To enable a speed-up of a realistic I-V calculation a combination of these parallel algorithms was implemented. Each bias point (I) can now be assigned to a group of CPUs, this group can be subdivided into different groups of momentum points (k), and these groups can be subdivided into groups of energy points (E). Four parallel schemes are therefore possible: I-k, I-E, k-E, and I-k-E. The user can specify the desired level of parallelism and the size of the groupings. An automated assignment of group sizes tries to select large parallel groups starting from the coarse level parallelism. **Figure 7** below compares parallelism in I-k and I-E to the parallelism in I. A significant improvement of the speed-up from 32 to 45 on 64 CPUs is shown. Some commensurability steps in the performance as a function of the number of CPUs are still visible suggesting the possibility of improvement on the automated CPU grouping algorithm.

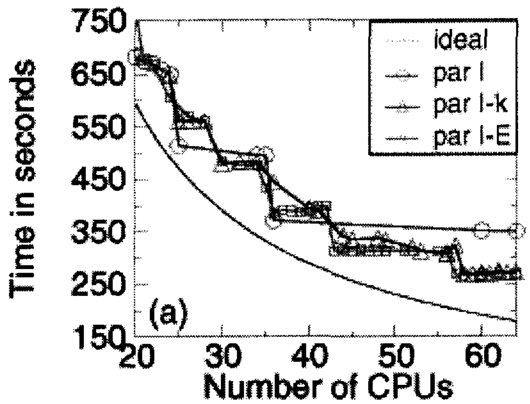


Figure 7A
(click for enlargement)

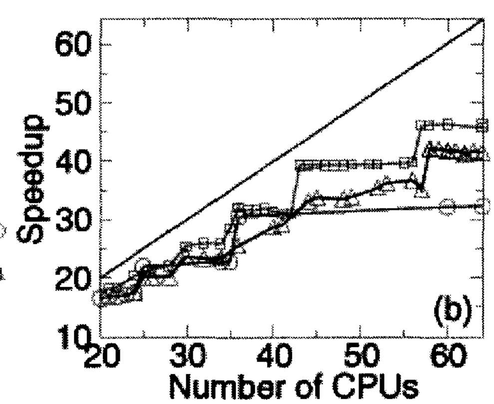


Figure 7B
(click for enlargement)

Significance

The numerical infrastructure built during the past three years provides the foundation for a CAD tool that will enable the development of nanometer scaled technologies for real world applications. A bottom-up philosophy has been adopted such that the tool can handle any nano-scaled structure comprised of covalently bonded atoms.

Status / Plans

With our currently developed technology we are now at the stage where we can proceed on production runs where we analyze Alloyed Disordered Systems and Interface interdiffusion Problems. Our ability to place atoms individually in a nanoelectronic structure allows us to look at alloyed materials such as AlGaAs, InGaAs or SiGe in an atomistic sense. In small structures the placement of individual atoms on the crystal is anticipated to have larger effects than on larger systems. We are planning to explore this dependence and understand the ultimate limits of optical linewidths due to alloy disorder. Several publications are in preparation.

The next step in the nanoelectronic modeling effort will be the development of software that enables the simulation of the interaction of a few electrons. Such interactions are important for the correct determination of optical bandgaps as well as for the exploration of charge coupled devices such as quantum cellular automata. We are planning to implement direct charge interactions as well as exchange interactions in a Slater determinant basis scheme. This scheme will be based on the representation of the relevant charge integrals on a single particle basis representation. We are also planning to incorporate algorithms that will enable the study of spinspin interactions in nanoelectronic systems.

Publications

The following list shows the details of our publication record of FY 00 due to full or partial funding of this nanoelectronic modeling task. The total count is 5 journal articles, 3 proceeding articles, 4 invited conference presentations, 9 contributed conference presentations, and 2 invited institutional seminars.

Peer-Reviewed Journal:

[1] Timothy B. Boykin, R. Chris Bowen, and Gerhard Klimeck, "Electromagnetic coupling and gauge invariance in the empirical tight-binding method", Physical Review B, Vol 63, pg. 245314 (2001).

[2] Gerhard Klimeck, R. Chris Bowen, and Timothy B. Boykin, "Strong wavevector dependence of hole transport in heterostructures", Superlattices and Microstructures, Vol 29, No. 3, pg. 187-216 (2001).

[3] Seungwon Lee, Lars Jönsson, John W. Wilkins, Garnett Bryant, and Gerhard Klimeck, "Electron-hole correlations in semiconductor quantum dots with tight-binding wave functions", Phys. Rev. B Vol. 63, 195318 (2001).

[4] Christian Rivas, Roger Lake, Gerhard Klimeck, William R. Frensley, Massimo V. Fischetti, Phillip E. Thompson, Sean L. Rommel, and Paul R. Berger, "Full Band Simulation of Indirect Phonon-Assisted Tunneling in a Silicon Tunnel Diode with Delta-Doped Contacts", Applied Physics Letters, Vol. 78, pg 814, (2001).

[5] Gerhard Klimeck, R. Chris Bowen, and Timothy B. Boykin, "Off Zone Center (Indirect Bandgap Like) Hole Transport in Heterostructures", Phys. Rev. B., Vol 63, pg. 195310 (2001)

Peer-Reviewed Proceedings:

[1] Gerhard Klimeck, "Indirect bandgap-like current flow in direct bandgap electron resonant tunneling diodes", Proceedings of the 2nd Workshop on Computational Materials and Electronics, Motorola University, Tempe, Arizona, Nov 9-10, (2000). Physica Status Solidi (b) Vol. 226, pg. 9-19 (2001)

[2] D. Keymeulen, Gerhard Klimeck, R. Zebulum, Adrian Stoica, Yili Jin, Carlos-Salazar Lazaro, "EHWPack: an Evolvable Hardware Environment using the Spice Simulator and the Field Programmable Transistor Array", In the Proceedings of ANNIE12000 (Smart Enginering System Design), St. Louis, MO, November 5-8, 2000

[3] Gerhard Klimeck, R. Chris Bowen, and Timothy B. Boykin, "Transverse Momentum Dependence of Electron and Hole Tunneling in a Full Band Tight-Binding Simulation", Proceedings of the 27th international Symposium on Compound Semiconductors (ISCS), IEEE, pg. 257 (2000)

Point of Contact

Gerhard Klimeck, Principal Member
Jet Propulsion Laboratory
gekco@jpl.nasa.gov
http://hpc.jpl.nasa.gov/PEP/gekco
Phone: (818) 354-2184

Additional Links

- Nanoelectronics at JPL
- NEMO-3D

- NEMO-1D

Page last modified on 12/26/01

Parallelization of the Regional Ocean Model System (ROMS)

FY2001 Annual Report

Objective

ROMS is a regional ocean general circulation model for solving free surface, hydrostatic, primitive equations over a varying topography. The shared-memory ROMS, developed at University of California at Los Angeles (UCLA), was based on the serial version of the S-Coordinate Rutgers University Model (SCRUM). It is a very useful tool for studying both complex coastal ocean problems and basin-scale ocean circulation. This is in contrast to other ocean models that are designed to study either coastal or basin-scale problems. The ROMS code can only be run on shared-memory systems. With the increasing need to simulate larger model domains with finer resolutions, there is a need from the ocean modeling community to have a ROMS code that can be run on any parallel computer with10s or even 100s processors. Our objective was to design a parallel MPI version of ROMS, and to keep minimum modifications of the code so that the original numerical algorithms remained unchanged, so that any user of ROMS could easily use this parallel version without any specific training in parallel computing.

Approach

To achieve this objective, we focused on the data structures of the code to discover all possible data dependencies. After investigating the entire package, the surface 2D computing domain was chosen as our candidate for parallelization. Based on domain decomposition techniques, and the MPI communication API (Application Programming Interface), a parallel MPI ROMS has been developed. In order to achieve load balancing and exploit maximal parallelism as much as possible, a general and portable parallel structure based on the domain decomposition techniques was designed for the flow domain, which has 1D and 2D partition features and can be chosen according to different geometries. MPI software is used for the internal communication, which is encountered when each sub-domain on each processor needs its neighbor's boundary data information. The module for communication is implemented separately from the main ROMS package, and can also be used for other sequential software with a similar data structure for parallelization. After various tests of the communication module, the combinations of BLOCK SENDS and UNBLOCK RECEIVES were used for data exchange, which provided the best results among other choices on the SGI Origin 2000 system. Besides communications among some internal grid points, the communication module was also required for boundary points communication if periodical boundary conditions were applied.

The implementation was carried out on the distributed memory systems, and the code currently runs well on the SGI Origin 2000. It can be easily ported to other parallel systems which support MPI.

Accomplishments

Teamed up with the JPL ocean group (Drs. Yi Chao and Y. Tony Song), we have successfully developed a parallel version of the ocean model ROMS on a shared-memory and distributed-memory system. An efficient, flexible, and portable parallel ROMS code has been designed using the MPI programming model. It provides significant speedup, which will dramatically reduce the total data processing time for complex ocean modeling. The code scales very well for certain numbers of processors, and achieves good speedup in performance as the number of processors increases. Although the code experienced a slow down for larger numbers of processors, it can be improved if optimization techniques are applied. The code is ready to be ported to any shared-memory or distributed-memory parallel systems which support the thread programming model or the MPI programming model. Based on the parallel version of ROMS, new numerical simulations of ocean flows have been carried out.

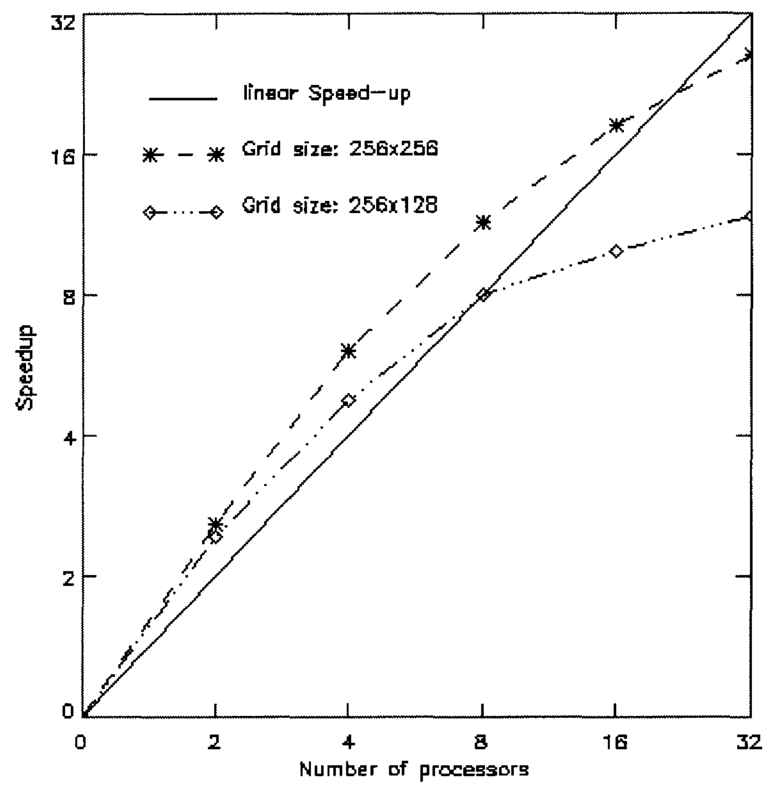
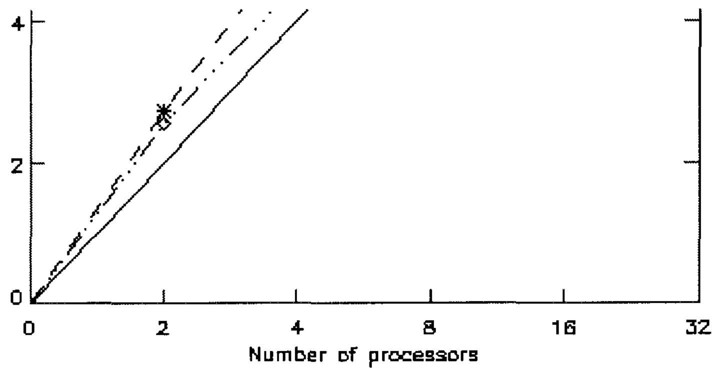


Figure 1. Speedup of the parallel MPI ROMS on the SGI Origin 2000 system with two different grid sizes.

Figure 1 shows the speedup of the parallel MPI ROMS with two different problem sizes. It gives excellent speedup vs. the number of processors. From the view of scalability, a large grid size problem gives better scaling results. Super-linear scalability is achieved on 16 processors for a problem with a grid size of 256x256, and the scaling results for a problem with a grid size of 256x128 are also excellent with a smaller number of processors. The usual explanation of super-linear speedup is that when more processors are involved, code fits into the cache better. Once size_of_the_problem becomes smaller than the Number of CPUs multiplied by cache size, the super-linear speedup ends and the usual communication overhead causes performance degradation. The SGI Origin 2000 is still one of the best scalable parallel systems for running a distributed-memory programming model which is suitable for running large-scale scientific applications with a large number of processors. The

speedup curves ha lue to the increase
of communication w



Significance

Ocean modeling pl conditions and predicting the future y sparse measurements ove cover the globe, it only provides inform ce has to be obtained from 3-dimensional ocean models.

ROMS solves the free surface, hydrostatic, primitive equations over varying topography using stretched terrain-following coordinates in the vertical, and orthogonal curvilinear coordinates in the horizontal. The object of this model is to study complex coastal ocean problems as well as the basin-scale ocean circulation. This is in sharp contrast with the more popular ocean models, such as the Modular Ocean Model (MOM) or Parallel Ocean Program (POP), which were primarily designed to model the basin-scale problems.

The parallel MPI ROMS has also been tested with various different problems, and the present results illustrated here clearly demonstrate the great potential for applying this approach to large-scale scientific applications.

Status/Plans

Based on the MPI version of ROMS, we have constructed an application on the North Pacific Ocean. The model domain extends in latitude from 45 degrees South to 65 degrees North, and in longitude from 100 degrees East to 68 degrees West. It is discretized into 384 by 224 grid cells, with a horizontal resolution of 0.5 degrees. The underlying bottom topography is extracted from ETOPO5, with a minimum depth of 20m near the coastal wall, and a maximum depth of 5500m in the deep ocean.

A snapshot of the simulated sea surface temperature at the end of the 10-year spinup is shown in Figure 2. There is a clockwise gyre in the north Pacific and a counterclockwise one in the south Pacific, with an equatorial current system between them. The entire equatorial current system is well and clearly developed, as shown by the tropical instability wave in the eastern equatorial Pacific.

Qualitatively, the MPI-ROMS reproduces many of the known features in the Pacific Ocean, a testimony to the success of the parallelization procedure. We are in the process of systematically evaluating the model solution against the existing knowledge and observations. In the future, we plan to run the MPI-ROMS at much higher spatial resolutions with an aim to simulate both the basin-scale and coastal processes.

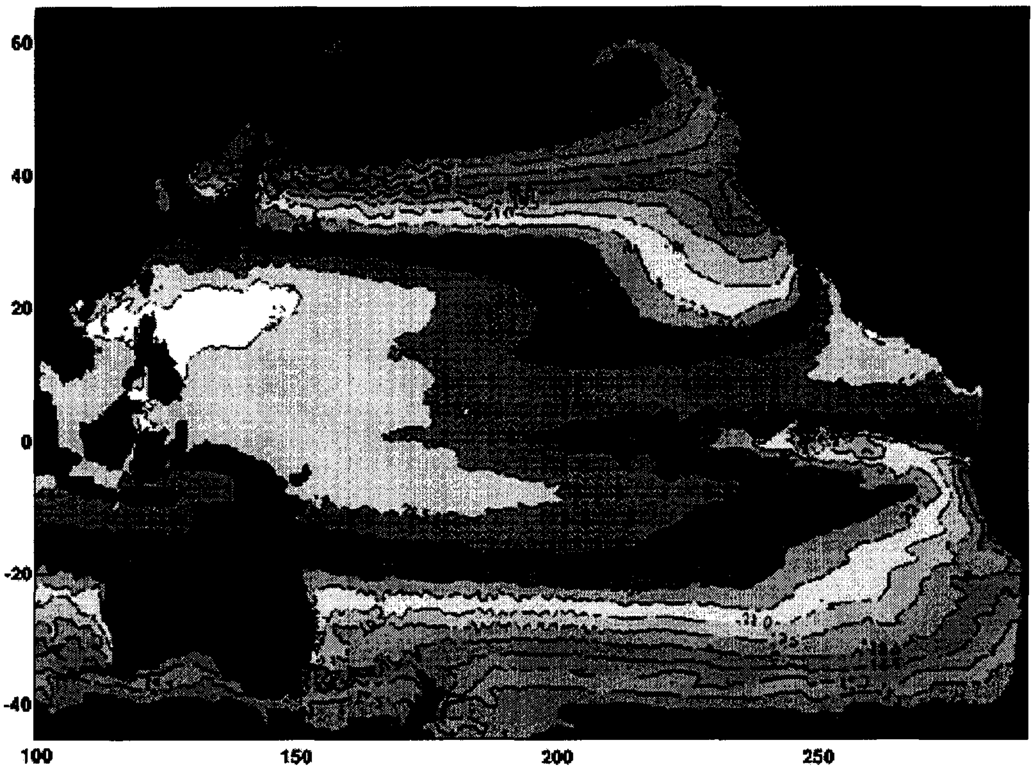


Figure 2. Snapshot of simulated sea surface temperature from the MPI-ROMS North Pacific model (Click to enlarge image)

More numerical tests and analysis with large grid sizes for such an application are underway. Final documentation of the parallel MPI ROMS needs to be completed.

Point of Contact

Ping Wang
Jet Propulsion Laboratory
wangp@azalea.jpl.nasa.gov
(818) 393-1941

Acknowledgments:
The author wishes to acknowledge Dr. Alexander F. Shchepetkin and Professor James C. McWilliams at UCLA for providing the original ROMS code and the help during the course of implementation of the parallel MPI ROMS.

Previous Report List Next

Page last modified on 12/26/01

Distributed In-situ Mission Simulation

FY2001 Annual Report

Objective

The goal of this task is to develop a simulation-based mission design environment that enables a collaborative mission operation synthesis among geographically distributed autonomous system design activities. This year's task objective was to develop a progressive automation framework that can provide a comprehensive verification and validation of various autonomous approaches including EDL (entry, descent, and landing) operation, rover operation, and in-situ science expedition. The framework involves site property knowledge engineering, virtual site construction, mobility platform and instrument property modeling, and instrument measurement simulation.

Approach

An In-situ site knowledge system (ISSK) was developed in three tiers. The first tier is composed of site property modeling and synthesis, for construction of a virtual in-situ environment. The second tier is composed of sensor system models for simulating science and engineering data products during an in-situ exploration. The third tier is the operation interface which supports both the mobility platform simulation clients (with the surface interaction), and the operation planning activity clients (with the observed site phenomena).

- Virtual site is represented with the physical properties relevant to surface interaction with simulated mobility platforms or observed phenomena by instrument systems. The virtual site simulates three types of properties: terrain, material, and texture.
- Measurement simulation integrates three aspects: photometric phenomena created by the Sun, geometric projection imposed by the sensor's viewing geometry, and instrument response governed by operation command and instrument system characteristics.
- Operation interface provides three types of simulation servers with a server-specific protocol defined in collaboration with the client research activities: 1) Descent Imager server with the EDL research team, 2) PanCam server with the vision-based research team, and 3) Site server with the rover dynamics simulation team.

Accomplishments

- Site Property Synthesis

The terrain property is described as a combination of a global surface elevation model and a set of local features. The global surface elevation model can either be artificially synthesized, based on the slope distribution and roughness specification, or derived from a morphology database. The surface material composition is described with a mixture type and mineral members. The mixture type indicates the composition ratio of minerals and can be specified as "LINEAR" (equal composition), "RANDOM" (a random composition) or "WEIGHTED" (a specific proportion). The virtual site regards the soil type information relevant to the surface interaction with a rover (e.g., slippage, sinkage) and the low resolution representations of certain terrain types (e.g., rock texture) as texture properties.

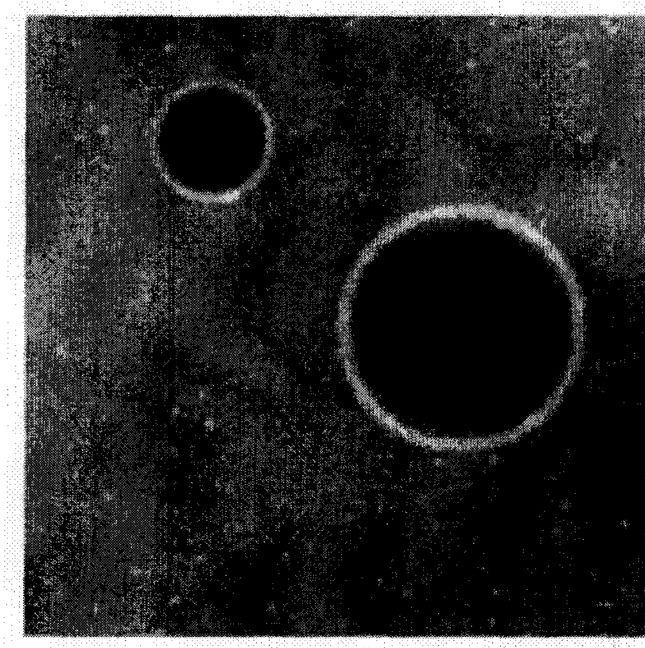


Figure 1

- Measurement Simulation

The photometric phenomena are simulated by creating a Sun object and a Sky object. The Sun object is created by specifying the distance and the direction from the site. The Sun object is passed to the Terrain object for a shadow mask and shading coefficient generation. Depending on the viewing geometry of the sensor, an appropriate geometric transformation is applied to the Terrain and Texture objects. The distortions due to various instrument response characteristics are implemented for spatial, spectral, and amplitude distortion types.

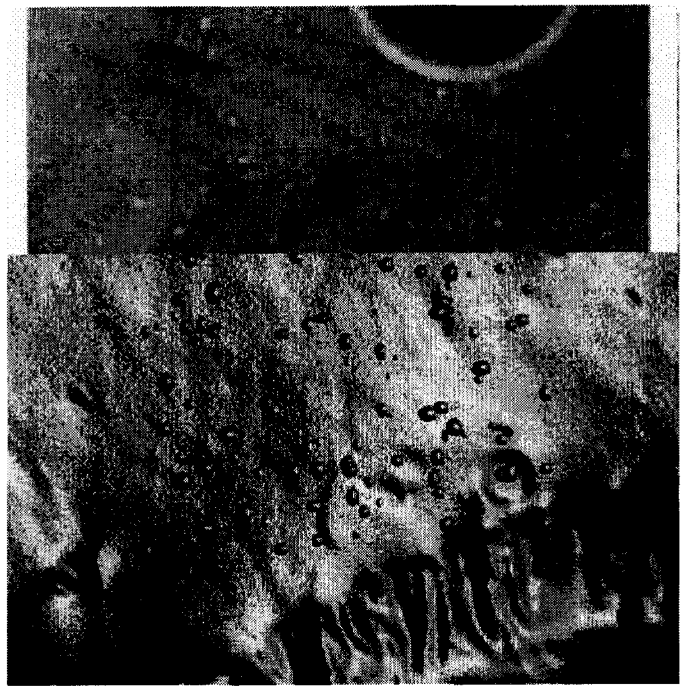


Figure 2

- Operation Interface

A descent camera image server and a panorama camera image server were created based on prototype camera models. The servers interact with respective clients for virtual site selection, instrument operation mode setting, sun direction setting, and provides simulated image frames.

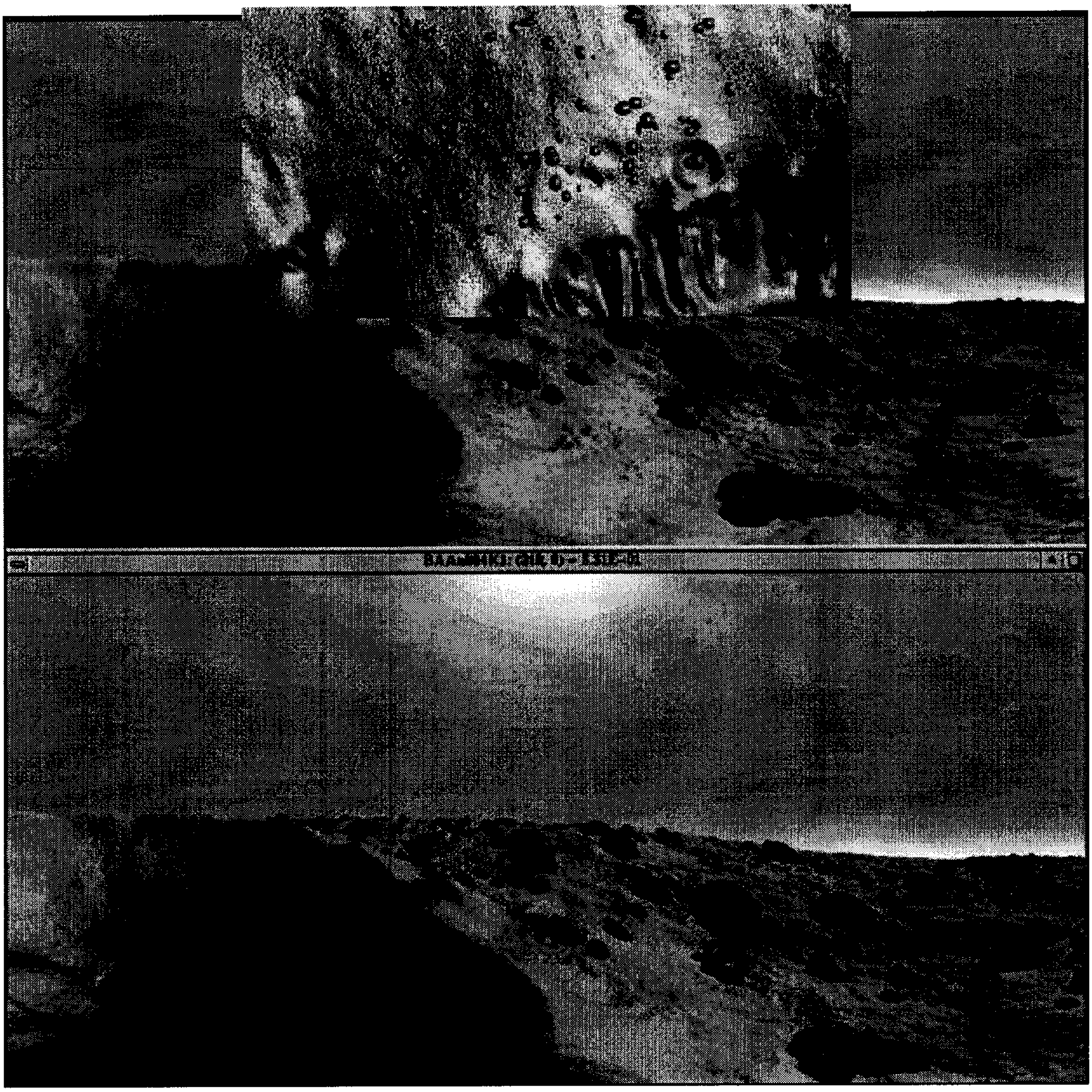


Figure 3
(click image to view enlargement)

Significance

Many research tasks have been devoted to enable controlled safe landing, and effective science exploration, with minimal human intervention. Along with the autonomy research, a corresponding level of verification and validation process research needs to be carried out so that the performance of an autonomous method can be accurately assessed prior to mission insertion. A comprehensive site property synthesis capability is of critical importance in developing a verification and validation framework for in-situ mission autonomy (autonomy testbed).

A virtual in-situ mission environment where multiple rovers and rover-based science instruments can be operated on a wide range of hypothetical operation sites is of critical importance in the operation planning and risk analysis of a mission. The framework enables collaboration among science experiment designers, instrument system developers, and rover operation teams. In particular, the framework can provide a comprehensive testbed for autonomous operation algorithm developers, both rover and instrument.

Sta

The and
meas
resea t
team in
supp

Poi

Meemong Lee
Jet Propulsion Laboratory
Meemong.Lee@jpl.nasa.gov
Phone: (818)354-2228

Publications

1. "Design-based Mission Operation", M. Lee, R. Weidner, W. Lu, IEEE Aerospace Conference, BigSky, Montana, March 2000.

2. "A Component-based Implementation of Agents and Brokers for Design Coordination," R. Weidner, IEEE Aerospace Conference, Big Sky, Montana, March 2001.

3. "In-Situ Site Knowledge System," M. Lee, submitted to IEEE-Aerospace Conference, BigSky, Montana, March 2002.

Previous | Report List | Next

Page last modified on 12/26/01

NASA's HPCC

ESS

EARTH & SPACE SCIENCES PROJECT

Mars Beowulf

FY2001 Annual Report

Objective

The FIDO Mars Exploration Rover (MER) relies on detailed panoramic views in its operation for several tasks:

- Determination of exact location
- Navigation
- Science target identification
- Mapping

Mars rover cameras gather many individual images with a resolution of 480x640 that are stitched together into a larger mosaic. Before the images can be stitched together they may have to be warped into the reference frame of the final mosaic, since the orientation and the individual images change from one to the next and since the final mosaic might be assembled in different views. The algorithm is such that for every pixel in the desired final mosaic a good corresponding point must be found in one or more of the original small images. This process depends strongly on a good camera model.

A good correlation of the individual pixels of stereo image pairs enable the construction of data that relates each pixel to its object position in three dimensions (x,y,z).

Current serial software implementations require 45 to 90 minutes computing time for the assembly of one mosaic or the correlation of two images. Approximately 100 image correlation pairs and a few different mosaics need to be processed within a one to two hour time window at the beginning of a ten hour operation window of communication to mars. It is our target to reduce that time by at least one order of magnitude from the original processing time to enable near-real time processing of the mars images.

Approach

We utilize our Beowulf clusters to parallelize the existing mosaicing and correlation processing software that was originally developed by the multi-mission image processing laboratory (MIPL) at JPL. The software will be parallelized using the standard message passing interface library (MPI) on two different clusters: the MIPL 20 CPU Pentium III 450 MHz cluster and the HPC Group 64 CPU Pentium III 800MHz cluster.

Figure 1. Mosaic Assembly Example

Accomplishments

The original algorithm takes about 90 minutes to complete a whole mosaic on a 450MHz Pentium III PC running Linux. It was desired to reduce this processing time down by at least an order of magnitude. We performed some algorithmic changes to the original software and we parallelized the existing mosaicing software on a Pentium III based beowulf cluster using MPI. A processing time from down to 1.5--6 from about 90 minutes was achieved (depending on the image and the processor speed). The images shown above were taken in a mars exploration rover (MER) test in the beginning of May 2001. The mosaic generation of this particular image (note that only about 1/4 of it is shown) took 3.3 minutes on 16 CPUs of a 450MHz Pentium III cluster.

Parallelization Algorithm:
The parallel algorithm divides the targeted mosaic into N slices, where N is the number of CPUs. Once each CPU has completed its tasks, it reports the image to the manager CPU, which patches the slices together into one image and saves it to disk.

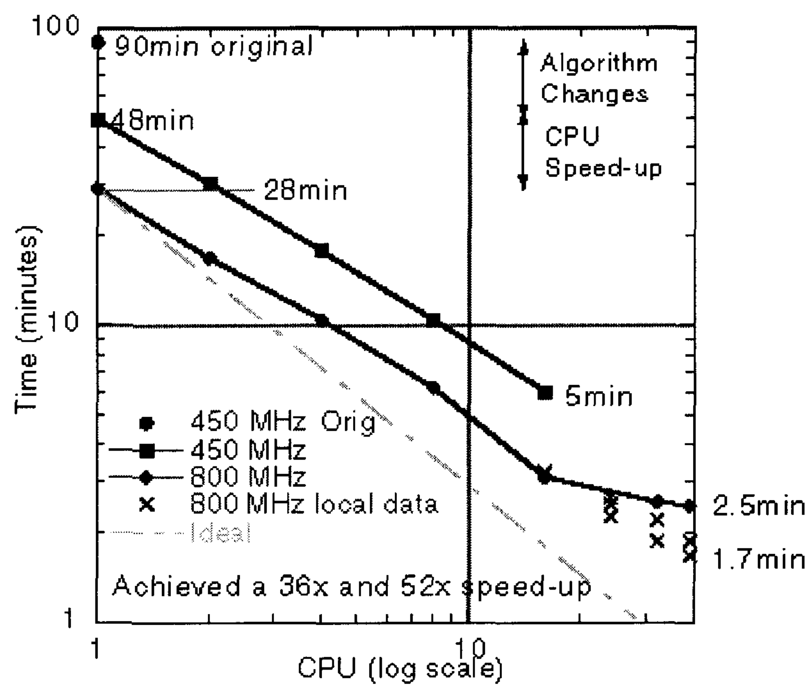


Figure 2. Execution time of the benchmarked mosaic process as a function of number of CPUs. Two different Pentium III clusters were used (450MHz and 800MHz).

Timing on one CPU:

The original algorithm took about 90 minutes on a single 450MHz Pentium III CPU to compose 134 images into a single mosaic. Algorithm optimizations result in a reduction of the required CPU time to about 48 minutes. Running the same algorithm and problem on an 800MHz CPU results in a time reduction to about 28 minutes.

Timing on multiple CPUs:

Parallelization of the mosaicing algorithm is shown (in **figure 2**) on the 800MHz cluster in a red line and on the 450MHz cluster in a blue line. The dot-dashed line in green shows the ideal speed-up. The actual timings follow the linear scaling quite nicely. Deviations from the ideality can be attributed to load balancing problems and data staging problems. The red line extends to a larger number of CPUs since the 800 MHz cluster has twice as many CPUs available. The parallel algorithm deteriorates strongly starting at 24 CPUs. This can be attributed to data staging problems to all the CPUs. If the images are copied to the local disks on each node of the cluster, the overall performance is significantly improved (purple crosses). The total processing time is reduced from 2.5 to 1.7 minutes. This comes at the expense of about 7minutes to copy the data to the local disks via rcp. This implies that if the algorithm is to be run on that many CPUs, a different method and / or hardware must be found.

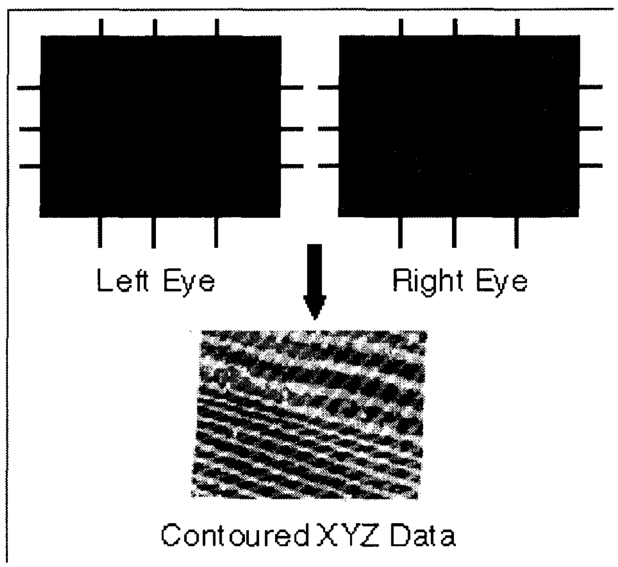


Figure 3.

Stereo Image Pair Correlation Results:

Two eyes in humans and animals provide depth perception. The images are obtained from two independent optical systems; the brain correlates the two, and provides a 3-D impression. The mars exploration rover carries several cameras including two dedicated sets of stereo cameras (navigation and panorama). The stereo images are used to determine distances to certain objects and the generation of 3-D maps. While the optical correlation in the brain appears to function with incredible ease, the digital correlation of two images is a numerically intensive task.

The stereo image correlation software attempts to deliver for every pixel (x,y) in the reference image (left or right) a corresponding pixel coordinate (x',y') in the corresponding pair (right or left). This is depicted below in **figure 4**, with a mapping from the red point to the green point. This is performed in a parallel algorithm that subdivides the reference image onto different CPUs.

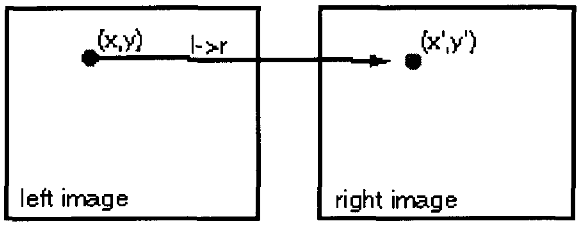


Figure 4.

The original algorithm was developed in the Multi-mission Image Processing Laboratory (MIPL). It consumed typically 90 minutes on a 450MHz Pentium III CPU running Linux. Under some conditions the algorithm was plagued by conversion problems, which could result in computation times of days or longer. This task was initiated to reduce the required time for image correlation by at least one order of magnitude by the usage of a Beowulf cluster. We performed algorithmic changes and parallelized the code. The new parallel algorithm partitions the original image into separate segments that are processed on different CPUs, as indicated by the blue lines in **figure 3** above. The new algorithm requires about 55 minutes on a single CPU (due to algorithm changes) and runs in about 4.5 minutes on 16 CPUs. Running the same problem on an 800MHz Pentium III cluster reduces the time further to about 3 minutes on 16 CPUs. Tests on old inputs that choked the original algorithm with conversion problems showed that the new algorithm appears to no longer be subject to these pathologies. The contoured XYZ data shown in **figure 3** is the essential input to the determination of distances to certain objects. The different color transitions indicate the transitions between a ridge and a valley behind that ridge (the valley is farther away than the ridge, resulting in a depth difference).

Significance

We have achieved the reduction of the required CPU time for the generation of a single mosaic and a single image pair correlation by more than an order of magnitude. This reduction of processing time will enable the full analysis of all mars data within a computation time of one to tow hours within a total window of communication to mars of 10 hours.

Status/Plans

The significantly reduced computation time will allow us to introduce new code that allows for error checking and correction in the correlation measurements. We are planning to develop this software now and integrate it into the data processing path. The software development so far has to be considered prototype development and the formal delivery of the software will require the proper software documentation.

Publications:

"Applications on High Performance Cluster Computers Production of Mars Panoramic Mosaic Images", Tom Cwik, Gerhard Klimeck, Myche McAuley, Bob Deen and Eric Dejong, Proceedings of the 2001 AMOS Technical Conference, September 10-14, Maui. (Reprint)

"The Use of Cluster Computers Systems for NASA/JPL Applications", Tom Cwik, Gerhard Klimeck, Charles Norton, Thomas Sterling, Frank Villegas and Ping Wang, Proceedings of AIAA Space 2001 Conference and Exposition Albuquerque, New Mexico 28-30 August 2001. (Reprint)

Points of Contact

Gerhard Klimeck, *Principal Member*
Jet Propulsion Laboratory
gekco@jpl.nasa.gov
http://hpc.jpl.nasa.gov/PEP/gekco
Phone: (818) 354-2184

Tom Cwik
Jet Propulsion Laboratory
cwik@jpl.nasa.gov
http://hpc.jpl.nasa.gov/PEP/cwik
Phone: (818) 354-4386

Acknowledgements

This work was also funded, in part, by the TMOD technology program under the Beowulf Application and Networking Environment (BANE) task. The original VICAR based software is maintained in the Multi-mission Image Processing Laboratory (MIPL). The work was performed in collaboration between Gerhard Klimeck, Myche McAuley, Tom Cwik, Bob Deen, and Eric DeJong.

Additional Links

- Near-Real-Time Mars Image Processing
- MIPL

Previous Report List Next

Page last modified on 12/26/01

NASA's HPCC
ESS
EARTH & SPACE SCIENCES PROJECT

Convection in Spherical Shells *[Thermal_Convection]*

FY2001 Annual Report

Objective

Our objective is to implement a general, well-optimized, parallel 3D time-dependent thermal convection software package in spherical shells on massively parallel supercomputers, and render it suitable for application to the broad range of problems in which the NASA community is involved.

Approach

The existing parallel implementation of the 3D thermal convection code, in a Cartesian coordinate system by P. Wang, was used as our basic model. We extended the code to a general software package for various applications in planetary sciences and geophysics. This was done (1) by allowing a range of geometric configurations to be included, (2) by including rotation, (3) by allowing a wider of variety of equations of state for the fluid to be included in the code, (4) by generalizing the code to solve anelastic equations, and (5) by designing a standard user interface module in a framework.

The discretization equations for spherical geometry are similar to those for rectilinear geometry; the coefficients in the current code do require some modification to reflect the spherical metric. Care is taken to write the equations in a flux-conservative manner in order to take full advantage of the finite volume method's property of exact global and local energy conservation. In addition, a provision has been made in the software to treat the periodic boundary conditions required for the longitudinal coordinate of the spherical system.

Efficient parallel implementation of the algorithms is of central importance to the overall performance of the model. Spherical geometry provides a greater challenge than rectilinear when achieving an optimal parallel partitioning of processors over the computational domain. For example, to implement a spherical geometry requires repartitioning the computing domain in a different manner than for the rectilinear box; one reason being the need for periodic boundary conditions in the longitudinal direction and for the "pole problems" in the latitudinal direction. We used the existing code's structure, but implemented new modules to deal with the cases that require special treatment due to the properties of their geometry. These modules include the formulation of a new linear system for the specific case, partitioning of the geometry for parallel computation, and implementation of new boundary conditions. With relatively minor modifications, the original multigrid solver was used for solving such linear systems.

Developing confidence in computational predictions essentially requires the establishment of quality assurance processes to govern modeling and simulation activities. The most common of these procedures were verification and validation. We addressed these issues at various levels. Once each module was fully tested, integration testing and comprehensive performance evaluation was performed along with correlation with theoretical performance estimates. Our final model was benchmarked by comparing with other published numerical results, and extensively validated against experimental data available in the recent literature.

Accomplishments

We have successfully implemented the finite volume method with an efficient and fast multigrid scheme to solve for three-dimensional, time-dependent, incompressible thermal convection flows in spherical shells on parallel systems. Preliminary numerical results were obtained for thermal convection problems in a spherical shell with an inner heating shell. Now the parallel software can solve various thermal convection problems with different boundary conditions, which include the Dirichlet, Neumann, and periodical conditions. An example is given in Figure 1, which displays the numerical results for the spherical convection problem with Rayleigh number 30,000, Taylor number 10x6, Prantle number 1.0, and a heating inner shell.

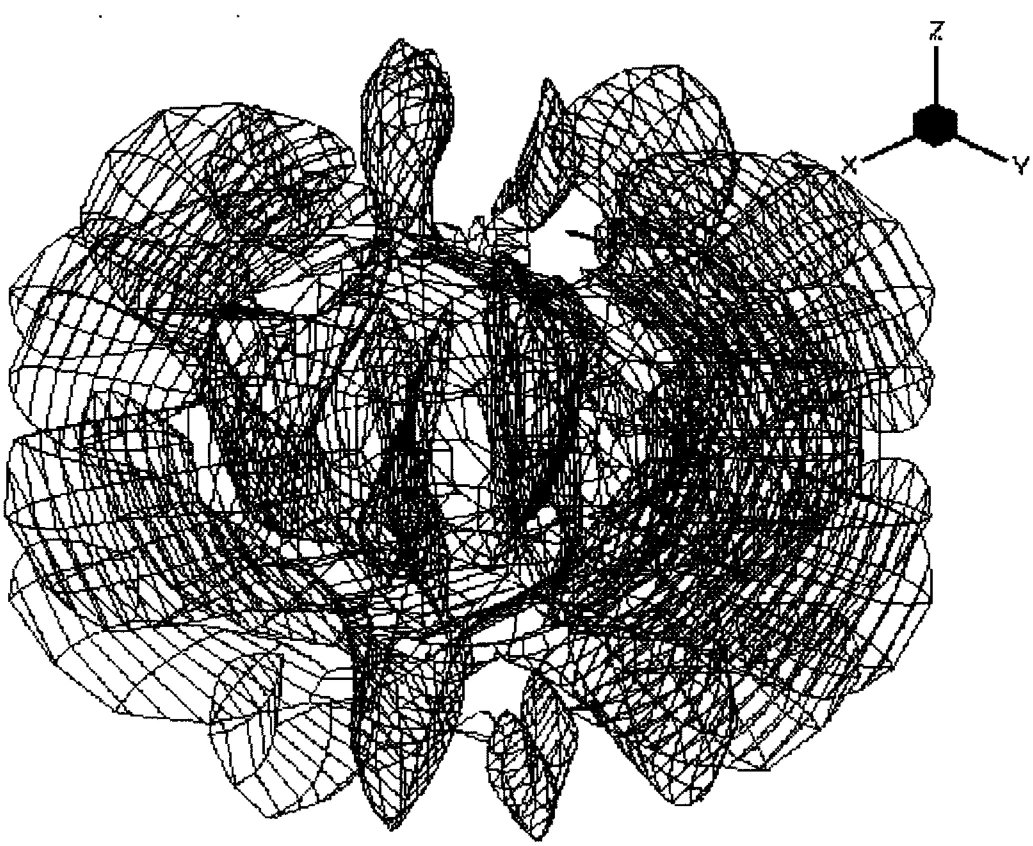
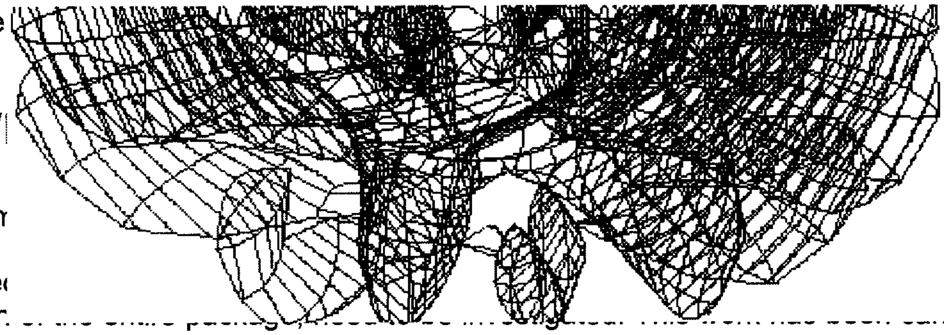


Figure 1. The vorticity field for the spherical convection problem.

Significance

Fluid motions driven by temperature gradients play a fundamental role in many important problems in planetary sciences and geophysics. Applications include the dynamics of atmospheres and oceans at synoptic and planetary scales, and convection in the earth's interior. Studies of the nature of thermal convection occurring in the Sun and Jupiter have been pursued vigorously for many years. Recent gains have been made in our understanding of solar convection through helioseismology and advances in supercomputing. Much less is known about convection in giant planets and brown dwarfs. Because extrasolar giant planets and brown dwarfs are now being discovered in large numbers, it becomes particularly important to explore how convection works in substellar objects in order to gain a better unde



Status/

Currently, n ... oordinate system are ... itions have been carrie ... , and optimization ... jointly by the author and Dr. Andrew James Friedson at JPL.

Point of Contact

Ping Wang
Jet Propulsion Laboratory
wangp@azalea.jpl.nasa.gov
Phone: (818) 393-1941

Previous Report List Next

Page last modified on 12/26/01

Evaluation of Gigabit Networking for Beowulf-Class Cluster Computing

FY2001 Annual Report

Objective

The High Performance Computing Systems and Applications Group was among the first to receive the new Myricom 2000 Switch with a peak performance rating of 2GBits/s. Our objective was to evaluate the functionality and benefits of Gigabit networking for improving the capabilities of Beowulf-Class cluster computers.

Approach

Our vendor, PSSC labs, installed the networking hardware. Almost immediately we realized that network performance was poor due to implementation issues in the underlying communication layer called GM as well as the MPI implementation. Myricom provided revisions to this software and we collaborated closely with them to detect network functionality and performance irregularities that affected our applications.

Accomplishments

By working with Myricom we have improved our network to near peak performance and characterized the communication performance of our PCI bus structure when shared-memory communication is used within dual-processor CPUs. Approximately 225 MBytes/s peak with 9.3 micro seconds latency can be achieved in Ping-Pong tests. In long messages, 130 MBytes/s with 1.5 usec latency (250 MBytes/s peak) can be achieved for shared-memory communication.

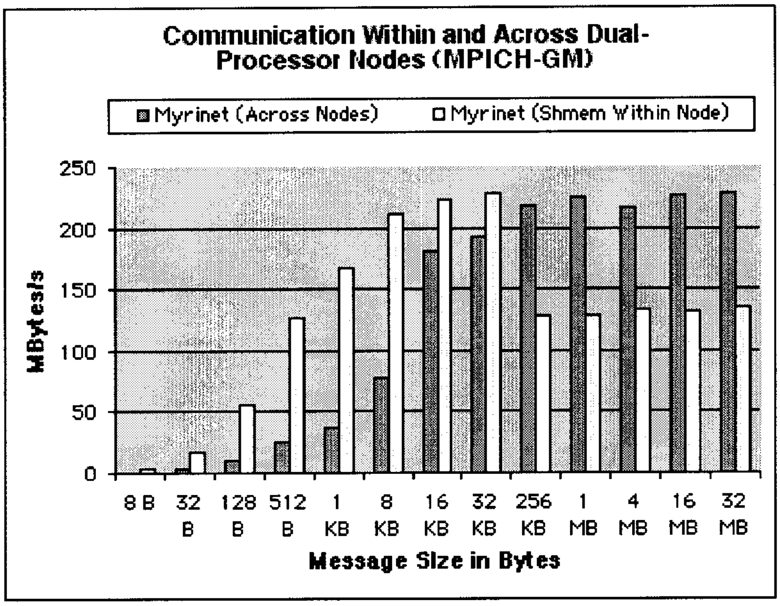


Figure 1. Network Performance Measurements Based on Internal PCI Communication with dual-Processor CPUs and Myricom's 2 Gbit/s Peak Switch Communication.

We have also demonstrated that Myrinet 2000 has much higher bandwidth than 100BaseT Ethernet, as expected, as well as the vendor specific network on the SGI Origin 2000. Incidentally, the Cray T3E network still outperforms Myrinet as it reaches peak speeds of 300 MBytes/s in identical tests.

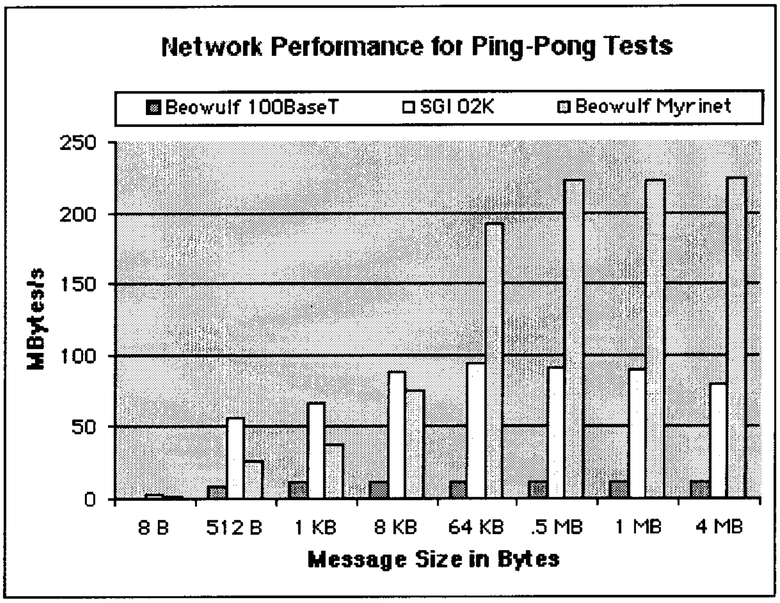


Figure 2. Network Comparisons for Myrinet, Ethernet, and Origin 2000 Cray-Link NUMA-based Communication.

In a separate study we have found that the network can positively impact large calculations that require large bandwidth communication, but the impact is not very significant if the number of processors used, or the message size, is small.

Significance

A significant effort was applied in working with the vendors to characterize and suggest recommendations leading to these performance improvements over our initial installation. Other Myrinet 2000 users worldwide are benefiting from these efforts. Indeed, scripts created from this work to support multiple compilers from vendors such as Absoft and Fujitsu are now part of the software distribution Myricom releases. Characterizing the impact of the new network using our PYRAMID adaptive mesh refinement library has also led to changes in the messaging software configuration. For example, shared-memory communication is no longer the default configuration for SMP processors as we have demonstrated that the PCI performance improvements and network bandwidth make this unnecessary-particularly for processors with limited cache sizes.

Status/Plans

We will continue to work with Myricom to examine the performance impact of the new network on our applications. We have found that the bisection bandwidth tends to drop as the number of communicating processors is increased so this one aspect of the network we will investigate further as many HPCC algorithms require high bisection bandwidth.

Points of Contact

Charles D. Norton
JPL Propulsion Laboratory
Charles.D.Norton@jpl.nasa.gov
Phone: 818-393-3920

Thomas A.Cwik
JPL Propulsion Laboratory
Thomas.A.Cwik@jpl.nasa.gov
Phone: 818-354-4386

References:

Charles D. Norton and Thomas A. Cwik. "Early Experiences with the Myricom 2000 Switch on an SMP Beowulf-Class Cluster for Unstructured Adaptive Meshing." In 2001 IEEE International Conference on Cluster Computing, Newport Beach, CA. October 8-11, pp. 7-14, 2001.

Previous Report List Next

Page last modified on 12/26/01

Interface Framework (WIGLAF)

FY2001 Annual Report

Objective

Current and future mission software will exploit the computational resources enabled by cluster computer systems [1]. Existing ground-based image processing software is now being ported to cluster systems to greatly reduce turn-around processing time for the 2003 MER rover mission operations, and production of science image products [2]. Areas of software validation, verification and model checking will use cluster resources to advance the process of building reliable software. Cluster systems are also used for software modeling and simulation for instrument design and visualization.

A cluster computer, a distributed system consisting of a possibly large number of nodes, is controlled remotely requiring data transfer and remote execution of parallel jobs. The goal of this work is to provide a web-based distributed graphical user interface framework that can be rapidly adapted to a broad range of applications that execute remotely on a cluster system. These applications may be written in any programming language and will execute on any cluster. The framework will enable rapid development of a graphical user interface for a specific application; control and monitoring of the application remotely executing on the cluster and data transfer. The interface will only require a web browser on the user's desktop, allowing it to be completely portable and available from any desktop location.

Pieces of a prototype framework (named WIGLAF) have been developed over the last few months. It should also be noted that though this framework is specifically designed for cluster systems, it is equally well suited for rapid interface development to applications executing on any remote system, including software controlling and collecting data from a remote test bed facility. The framework is also equally applicable to applications that execute on the user's desktop machine. In this case, no remote execution or data transfer is necessary since WIGLAF is used to rapidly generate application user interfaces.

Approach

WIGLAF development exploits the recent maturation of web-driven software technologies and frameworks for web-portals. The technologies are open source, readily available and driven by a large development and application community. Using these technologies allows portability and ease of use for both the application developer and end user.

The framework is deployed as a client-server architecture. The server executes on the cluster system, controls the application, and serves data to the client. The client is an applet that runs in a web browser, connected to the server host machine by a network. Key technologies include:

use of the Extensible Markup Language (XML) to hold all data associated with the application software [3], Java to provide a platform-independent execution environment on the cluster system and the graphical user interface generator displayed through the browser, and the Java Remote Method Invocation (RMI) package to provide simple, effective client-server networking. It is the rich interaction between XML and Java that enables graphical objects to be rapidly developed and constructed into graphical user interfaces specific to the application. An XML grammar is being built to hold the application data that will deploy the user's graphical objects. The grammar will be extensible to handle future applications.

XML for Data Integration
XML provides the structure and grammar to hold application data and to enable communication between the application server running on the cluster and the user interface available through the browser on the client [4]. XML provides a platform-independent data format for files that describe application input parameters, including internal and external documentation and output file locations. XML is a method for putting structured, hierarchical data into a text file. The text file's appearance is similar to HTML, but XML is not derived from HTML. Although the data is represented solely as text, an XML file is not meant to be human-readable.

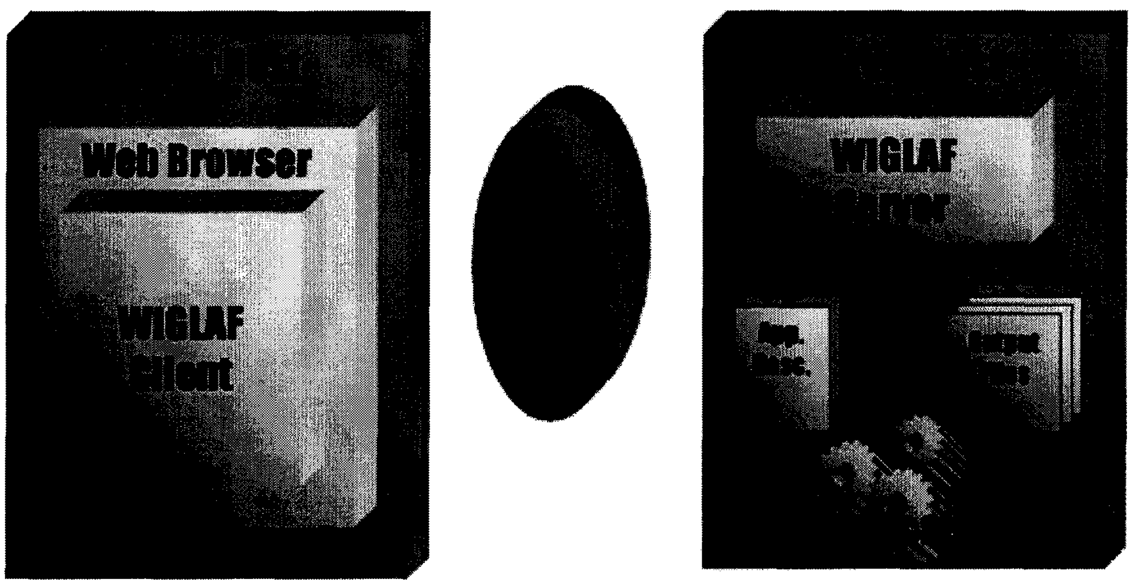


Figure 1. Schematic of existing WIGLAF architecture. The client-server architecture is enabled through Java RMI. The server executes on the cluster, controlling the application and data transfer to the client running in a web browser on the client host.

An XML file, called an application description file, is associated with each application. It contains input parameters, output file locations, documentation, and application execution instructions. It contains sufficient information to automate the generation of a GUI for the application, although the description is implicit. The application description file includes "meta-data" about each input parameter: name, type, description, grouping, default value, etc. It does *not* specify explicit GUI objects or options – this is left to the WIGLAF client. The client uses this meta-data to intelligently construct a GUI tailored to each application. The server uses this meta-data to execute the application with the appropriate input parameters and collect the output. This application description file is the enabling feature of the framework.

To date, we have begun to interface WIGLAF with a set of existing applications, and we have found that XML provides more than enough functionality to facilitate integration.

Java for Platform Independence Control
Java provides WIGLAF a platform-independent execution environment with a rich standard library, GUI application framework, and web-browser embedded applets. Java is an

object-oriented, platform-independent, dynamic language.

The Remote Method Invocation (RMI) package is used to provide high-level client-server networking. RMI is essentially an object-oriented Remote Procedure Call (RPC) mechanism for Java (RPC can be used with Java applications, but RMI is not built on RPC). Like RPC, it is not as efficient as socket-level network programming, but the conceptual simplicity and ease of development more than compensate for this shortcoming.

By using RMI networking, the application is network-enabled, not network-bound. While the application client and server can be run as separate processes on the same computer, they may also be run as a single, stand-alone application which does not require any network connectivity. This functionality was achieved with minimal effort and almost no additional code. As outlined in the introduction, this functionality allows WIGLAF to build interfaces to applications that execute on the user's desktop where no connectivity is required.

To simplify the development, the Ant build package is used. Ant is a product of the Apache Jakarta project, intended to provide a platform-independent version of the "make" build tool more suitable for Java software development [5]. The WIGLAF build system relies upon Ant, and can therefore be built on any platform that can run its server.

Accomplishments

- An informal XML document type definition exists (formally defined, but not enforced), which allows creation of customizable application descriptions.
- Client and server implementations exist which allow remote control and output visualization of applications via a web browser.
- Integration with an existing application has been accomplished without any changes to the application itself.

Significance

Several cluster computer systems have been developed across the laboratory and more are either being considered or are under development. The work funded under the Earth and Space Science (ESS) Project has demonstrated the cost effectiveness and performance improvements of applications executing on cluster systems. WIGLAF substantially answers one aspect of the 'ease-of-use' question regarding user access to cluster systems. Besides the Lab's currently available cluster systems, the Tropospheric Emission Spectrometer (TES) ground data processing system is currently designing a large computational facility that will likely consist of an (order) 100 CPU cluster system. WIGLAF can be used for rapidly developing interfaces for use with prototype code running on testbeds under consideration for the project. The Multi-mission Image Processing Laboratory (MIPL) is developing software for use with clusters for insertion into the 2003 MER mission for image processing. WIGLAF would allow broad access to these systems from both science team members as well as operations team members. The NGST project is currently developing interfaces to testbeds for control of optical systems at JPL, GSFC and at contractor sites. WIGLAF would allow rapid development of interfaces for these testbeds as well as existing software for modeling and design of the NGST optical systems.

Applications developers can leverage the WIGLAF GUI framework to provide an interface for new and existing applications. The addition of a WIGLAF GUI to a design optimization application required no changes to the application itself. A translator was written which produced input files formatted as the application expects. While the translator is not fully generalized (it cannot produce binary files), this is a lucrative integration path for pre-existing applications or applications that are difficult to modify.

New applications, or those that may be easily modified, can directly parse the XML application description file. This is the preferred integration method, because the application developer need not maintain a separate translator program. There is an existing Java library that simplifies access to these application description files, and Fortran 77, Fortran 90, C, and C++ bindings are also being developed. In many cases, graphical output plotting can be adapted with little extra effort.

Status/Plans

A prototype application interface framework exists, and has been used to provide a distributed GUI for a test application and a design optimization application. The prototype consists of a set of components including:

- A server that executes on the front-end of the cluster Nimrod running the Java Virtual machine under Linux. The server controls the application, publishes the XML application description file, monitors and caches the application output files, and publishes the application output files.
- An XML grammar that holds input and output data for a set of sample applications. This data includes a range of hierarchical grouped parameter types such as floats, integers, character strings, file names, etc. It also includes documentation about these parameters.
- A browser-based client that automatically generates a GUI from the XML application description file. This allows the user to enter input parameters to the application and visualize output. Output visualization is limited to text and two-dimensional plots.

Point of Contact

Tom Cwik
Jet Propulsion Laboratory
cwik@jpl.nasa.gov
Phone: 818-354-4386

Ed Vinyard
Jet Propulsion Laboratory
Edward.S.Vinyard@jpl.nasa.gov
Phone: 818-393-2548

References

[1] William W. Hargrove, Forrest M. Hoffman and Thomas Sterling, The Do-It-Yourself Supercomputer, Scientific American, pp. 72-79, August 2001

[2] Tom Cwik, Gerhard Klimeck, Myche McAuley, Bob Deen and Eric Dejong, Applications On High Performance Cluster Computers: Production of Mars Panoramic Mosaic Images, The 2001 AMOS Technical Conference, Waleia, Maui, September 10–14, 2001

[3] Normal Walsh, A Technical Introduction to XML; http://www.xml.com/pub/a/98/10/guide0.html

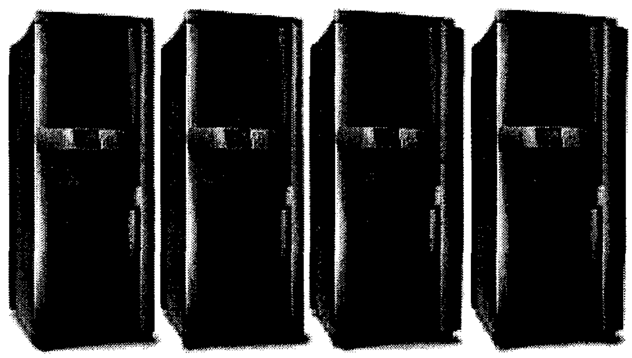
[4] Sun Microsystems, Using XML and Java technology to Develop Web Services: Keeping Migration Paths Open; http://www.sun.com/software/xml/cover/2001-0205/

[5] Steve Loughran, Ant in Anger: Using Ant in Production Development Systems; http://jakarta.apache.org/ant/ant_in_anger.html

Page last modified on 12/26/01

SGI Origin 2000 Update

FY2001 Annual Report



Objective

To continue to improve the SGI Origin 2000 hardware and software environment in support of expanding the number of applications in the Massively Parallel Processing (MPP) arena.

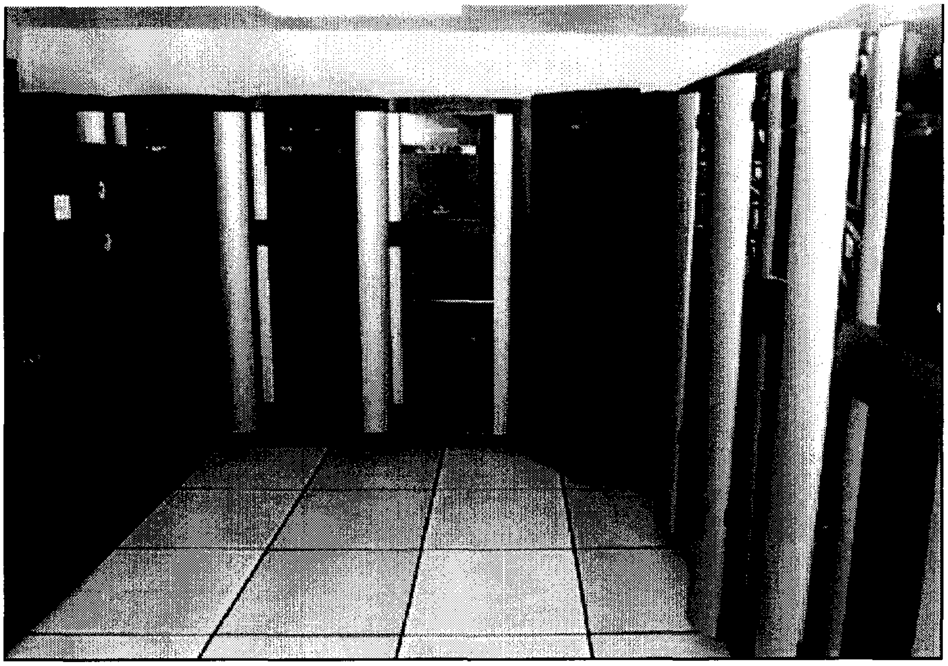
Approach

JPL has continued to fully utilize the SGI Origin 2000 supercomputer for Earth and Space Science projects. The SGI Origin 2000 is located in the Supercomputing and Visualization facility, Building 126, at JPL. The SGI Origin 2000 and the CRAY SV1-1A share a StorageTek Powderhorn tape storage system. This system was procured under a collaborative agreement with JPL institutional funds, JPL Ocean Science funds, and the NASA sponsor of the ESS project under NASA's HPCC Program.

Accomplishments

The installed hardware consists of 128 processors and includes:

- R12000, 300 MHz processors
- 64 GB of Memory
- 6 TB of Disk Storage



The basic element of the SGI Origin 2000 is the R12000, 300 MHz processors. All 128 processors are routed through an SGI MetaRouter bringing the entire system bandwidth to 20GB/sec (peak). The Origin 2000 runs under the UNIX-based IRIX 6.5.X, which has all the features normally associated with UNIX. Current software includes Fortran 90, C, and C++ compliers, as well as math and communications libraries, debugging tools, performance analysis and monitoring tools

Over the past year, utilization of the SGI Origin 2000 has been between 70% to 90% of the available cycles. This high utilization was made possible in part by the introduction of an improved batch queuing program that made efficient use of the CPU cycles

Significance

JPL will provide feedback to the SGI High performance Computing Division of its experience using the SGI Origin 2000 to enhance its hardware/software. Thus contributing to maintaining U.S. leadership in supercomputing. The applications running on the SGI Origin 2000 will help to address the Grand Challenges of the Earth and Space Sciences in the analysis of the enormous datasets from NASA's Earth and Planetary Missions.

Status/Plans

Currently, there are 239 NASA users in 36 projects. The system supports both shared-memory programs and message-passing programs for up to 128 processors.
Efforts in collaboration with SGI and JPL users include:

- Implementation of PBSPro software to further improve scheduling of batch jobs.
- Enhancing the management of the system by implementing a small front-end machine.
- Continuing to improve the way science and engineering data can be evaluated by using the SGI visualization capabilities.

Point of Contact

Larry Eversole
Jet Propulsion Laboratory
Larry.C.Eversole@jpl.nasa.gov
Phone: (818) 354-2786

Previous Report List Next

Page last modified on 12/26/01

PYRAMID Parallel Unstructured AMR Library

FY2001 Annual Report

Objective

Our objective is to build an advanced software library supporting parallel unstructured adaptive mesh refinement for large-scale adaptive scientific and engineering computations. The PYRAMID library combines a modern, simple, efficient, and scalable approach to meet this objective while leveraging the advantages of finite element methods beneficial to the scientific community. This year, our goal was to significantly improve the performance of time consuming parts of the library and to stress its functionality by working with much larger meshes containing millions of elements.

Approach

We acquired large meshes (in use by our collaborators) and analyzed the performance of various code segments. This process illustrated that certain mesh migration operations performed very poorly, even though the existing algorithms minimized the volume of the messages transferred. Porting and running the software on a Macintosh cluster using a version of MPI that measures data transfer rates showed that messages would often stall in the network due to the irregular nature of this problem. We developed a new mesh migration algorithm that is based largely on historic hypercube routing schemes, but with many enhancements. This significantly improved performance in mesh migration even though the volume of data transferred is larger than the minimal direct data placement schemes we had used previously.

Accomplishments

The modified library now supports adaptive refinement of meshes with many millions of elements much more efficiently. Element redistribution of this artery mesh segment previously required about 50 minutes. With the new algorithms in place this was reduced significantly to only 3 minutes. Similar communication modifications, that also increase reliability, are in progress in the adaptive refinement sections of the software.

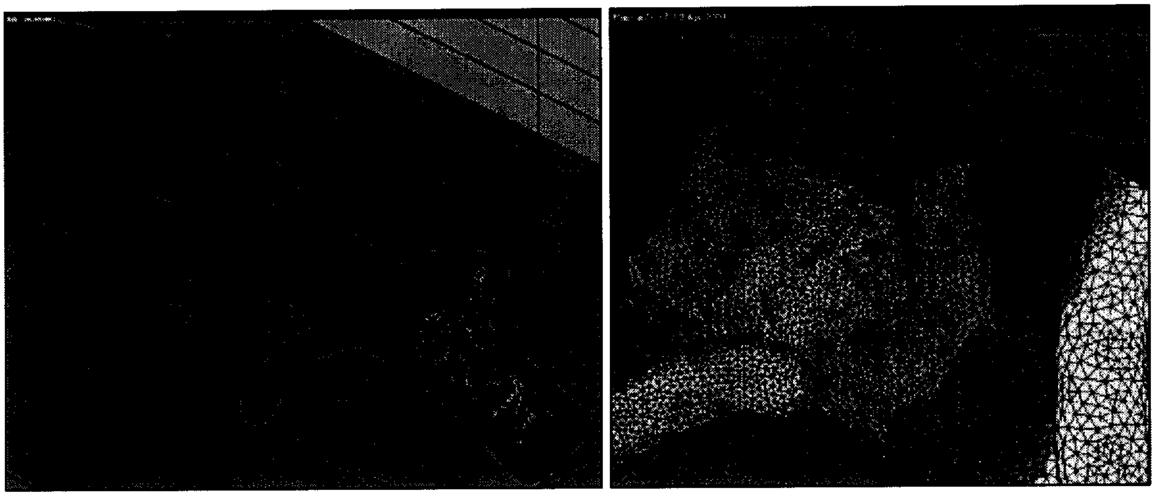


Figure 1. Adaptive Refinement Applied to Artery Mesh Example
(Click image to enlarge)

Significance

The communication performance improvements have impacted the usefulness of the library. Much of the development has been performed on our Beowulf cluster using Myrinet 2000 with 2 GBits/s peak performance, but the improvements are significant for 100BaseT Ethernet systems as well.

Status/Plans

We have recently established collaboration with Sandia National Laboratories to move development forward in support of large-scale mesh generation based, on meshes with initial coarse geometry. This, in addition to supporting ESS Round III Grand Challenge teams interested in using the library, will also focus our efforts toward adding new capabilities beneficial to their needs.

Point of Contact

Charles D. Norton
JPL Propulsion Laboratory
Charles.D.Norton@jpl.nasa.gov
Phone: 818-393-3920

References:

Charles D. Norton, John. Z. Lou, and Thomas A. Cwik. "Status and Directions for the PYRAMID Parallel Unstructured Adaptive Mesh Refinement Library." In 8th International Workshop on Solving Irregularly Structured Problems in Parallel, San Francisco, CA. April 23-27, 2001.

Previous Report List Next

NASA's HPCC

ESS

EARTH & SPACE SCIENCES PROJECT

Remote Interactive Visualization and Analysis of Earth and Planetary Image and Terrain Dataset

FY2001 Annual Report

Objective

RIVA (Remote Interactive Visualization and Analysis) System was developed and implemented on the Cray T3D back in 1995 under the sponsorship of the ESS program. It has been used in house as a tool to design and produce high-resolution fly-by movies using the supercomputing facility at JPL. The RIVA movies have been used both internally and externally to JPL for outreach and educational purposes.

Despite the popular demand for 3D fly-by movies, there has not been any funding for the development and enhancement of RIVA software since 1995. The software was built on obsolete hardware, such as the Cray T3D and HiPPI network, and proprietary software, e.g., Cray's Shmem library. The objective of this year's task is to port RIVA to SGI Origin 2000 using a standard communication API, and to enhance its functionality to be more robust and user-friendly, and lastly, to deliver it to the HPCC community with complete documentation.

Approach

RIVA is a parallel terrain rendering system that takes large earth or planetary images wrapped around digital elevation models to produce 3D perspective views. The core algorithm of RIVA is a *feed forward sort-last* parallel renderer called *"Ray-Identification"* algorithm. It uses both input data decomposition and output image decomposition to achieve parallelism and load balancing. It uses a spherical data model and is thus able to render global datasets such as Mars global mosaic (shown in **Figure 2**) or a 2-D surface dataset (such as the Lewis & Clark trail in **Figure 3**).

RIVA is an interactive visualization system that allows users to navigate a low-resolution dataset on their local workstation while having the high-resolution images rendered on the parallel supercomputer. The high-res. images will be displayed on user's workstation as they are rendered. The RIVA GUI, called *Flexible Flyer,* communicates with the renderer through a routing program called *Router.* Router is a message delivery service that supports point-to-point and multi-cast communication using abstracted message types and symbolic mailboxes. It hides the physical locations of the senders and receivers from the programmers, and it provides a reliable connection that does not break even if the sender or the receiver dies and restarts again. Flexible Flyer not only sends viewpoints to the renderer, but also a set of commands to control the rendering parameters, such as elevation exaggeration, opacity, or color mapping. An interface program, *hfb_host,* is used to process and relay the messages from the Flexible Flyer to the renderer.

RIVA is also an animation tool. The Flexible Flyer has a built-in key frame editor. A set of key frames can be selected using the navigator and a flight path is then generated using a cubic spline algorithm. The flight path can be edited and previewed in the Flexible Flyer before the final rendering. A 2-D map display program, xshow, can be used in concert with the key frame editor to fine tune the flight path. The flight path is then saved in a file and rendered later in batch mode. The key components in RIVA system and their inter-relationship are shown in **Figure 1**.

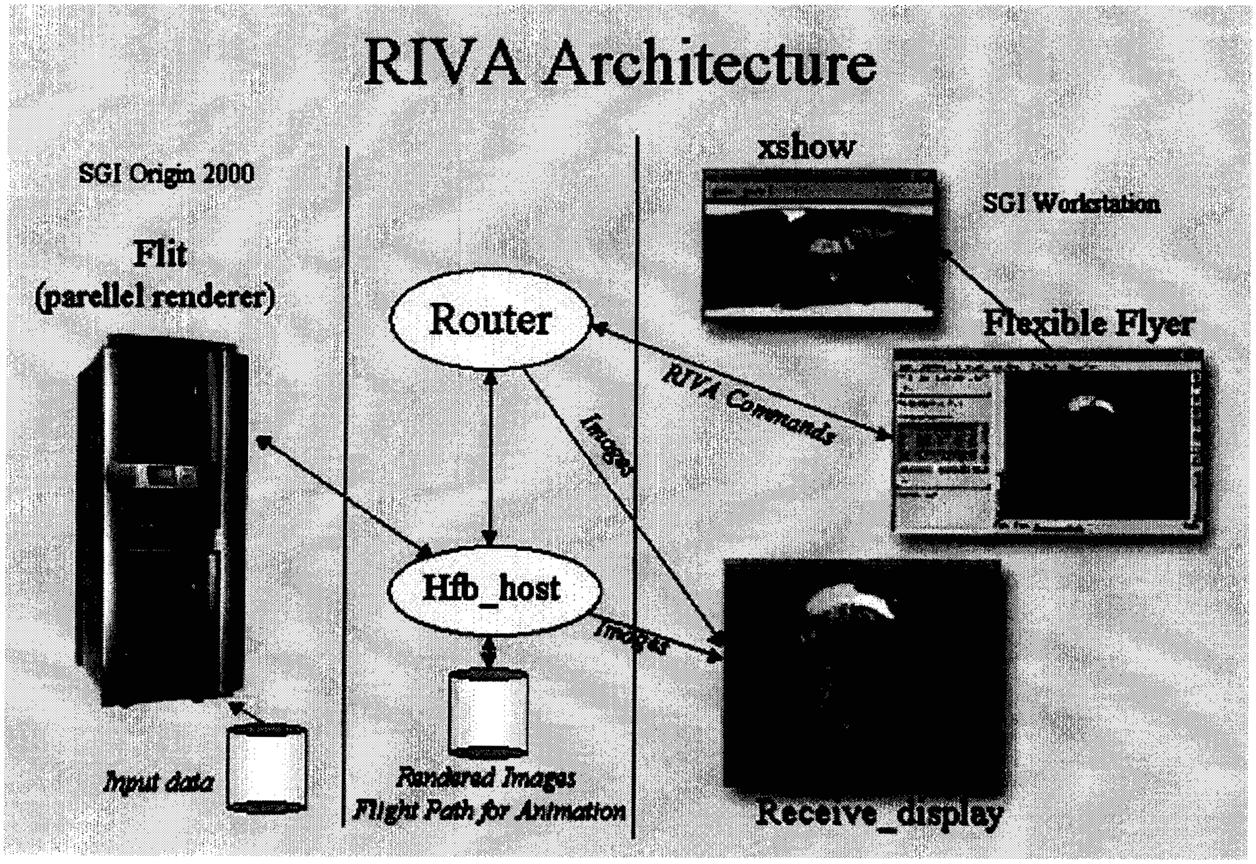


Figure 1. RIVA Architecture

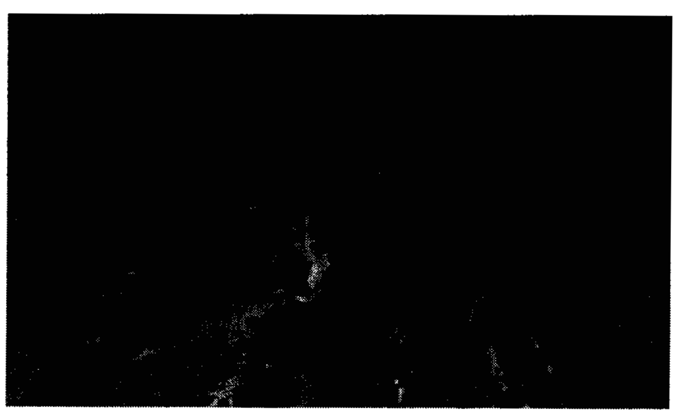


Figure 2. Mars Valles Marineris

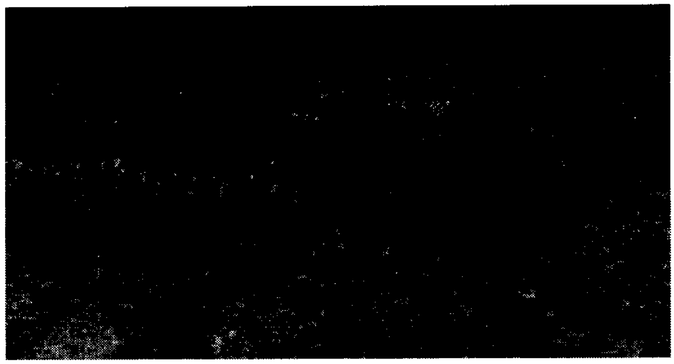


Figure 3. Traveler's Rest along Lewis & Clark Trail

Accomplishments

We have accomplished three milestones in FY2001:

1. Improved the flight path editing capability and RIVA GUI functions. Two programs were implemented to aid in the design of the flight path and the display of the rendered images. Xshow is a 2-D display program that displays the same terrain image used in the Flexible Flyer and the viewpoints generated by the Flexible Flyer. It displays the viewpoints either as key frames or as a continuous flight path. By viewing the key frames in the 2-D map a user can orient himself in a global sense and be able to space the key frames equally and smoothly. In addition, new features are added into Flexible Flyer to preview, pause, and resume the flight path, as well as display the current key frame's position (longitude, latitude, altitude and the viewing angle). They are all useful tools for fine-tuning the flight path. Receive_display is the second program we built last year to receive and display the rendered images on the local workstation. The program can receive the images either via socket connection to hfb_host, or via the Router. The Router connection is more flexible and reliable, but slower. The socket connection is faster, but will be lost when either side of the socket dies (for any reason). Together the three GUI programs xshow, receive_display, and Flexible Flyer, form the RIVA GUI for both remote visualization and animation design.

2. Ported RIVA to SGI Origin 2000. RIVA was originally written using Cray's Shmem library on the Cray T3D. It was first ported to the SGI Origin 2000 using SGI's shmem library, available in SGI's Message-Passing Toolkit (MPT). The goal for this year is to port RIVA to another type of message passing API that is not Cray/SGI proprietary, thus more portable to other hardware platforms. SVR4 UNIX shared memory library was chosen as the interprocess communication API. The reasons for choosing a shared memory library instead of a message passing library, such as MPI, are two fold: 1) RIVA was originally built using one-sided communication, it is a more straightforward port from shmem API to another shared memory library, and 2) RIVA uses SGI's Image Library (IL) to read in input files in non-RIVA file format. RIVA uses multi-thread non-synchronous read functions in IL to speed up the input operation. If MPI were used to distribute the input data, the data distribution cannot begin until all the data is read into memory, therefore, using shared memory for data distribution will be more efficient. Interprocessor communication is required in input data distribution and the final image composition. Reading an input file using IL library is about 10% to 20% faster than reading a RIVA file. Final image composition is less than 5% of the total rendering time while using 32 or less nodes. The performance difference between shmem and SVR4 shared memory API is insignificant.

3. Delivered RIVA 1.0 to JPL ESS Software Catalog. Running RIVA interactively requires up to six programs to run simultaneously, and some of these must be started in a specific order. To simplify the startup process, two runtime scripts were prepared, one for the renderer and the other for the GUI. To prepare for the release, we cleaned up obsolete code, streamlined the make process, and prepared complete documentation (including the installation guide, the tutorial, the main page for each program, the RivaFile document, and the Command Language Document). We also prepared a separate tar file for a test dataset. A New Technology Report (NTR) was filed for RIVA 1.0 and the tar files are made. It will be put in the Software Catalog when the NTR is approved.

Significance

RIVA has been used to generate many fly-over movies for earth and planetary datasets. In the past year, we produced a HDTV-format San Diego fly-over movie to be used in a Discovery Channel program introducing the geological evolution of San Diego. We also produced a movie of the Lewis & Clark Trail using a new 200 Gbyte LandSat mosaic of the US. RIVA can be used to render time-varying simulation data as well. An example is the North Atlantic Ocean Surface Temperature animation we produced a few years ago. RIVA is an efficient and robust software rendering system for terrain datasets; it is good for both interactive visualization and animation design and production. It is capable of handling very large datasets and rendering very large images without being limited by the hardware capacity. By making it publicly available, we hope people can use it to help themselves as well as the general public to better understand their datasets.

Status/Plans

The current functionalities in RIVA were mostly driven by the various demands of making animations. In the next year, we plan to apply RIVA and ParVox (a 3-D Volume Rendering system developed under the ESS program) as visualization tools for the ESS Round-3 Grand Challenge Teams. New functions will be added when they are identified for specific applications. A possible application is to visualize 3-D time-varying simulation results from the finite element software that models active tectonics and earthquake processes. One example is to use RIVA to produce animation of "surface wave" during the earthquake simulation, by combining elevation displacement extracted from a 3-D model to the high-resolution LandSat image and DEM model.

Point of Contact

P. Peggy Li
Jet Propulsion Laboratory
Peggy.Li@jpl.nasa.gov
Phone: (818)354-1341

Additional Links

http://alphabits.jpl.nasa.gov/RIVA/

Previous Report List Next

Page last modified on 12/26/01

Information Power Grid Applications Technology Development Task

FY2001 Annual Report

Objective

The Information Power Grid (IPG) is NASA's high performance computational grid. Computational grids are persistent networked environments that integrate geographically distributed supercomputers, large databases, and high end instruments. These resources are managed by diverse organizations in widespread locations, and shared by researchers from many different institutions. While the adoption of the Grid paradigm by working IT professionals has been slow, there are now ample signs that the maturity level has reached that needed to become an accepted infrastructure for the most demanding distributed applications. Two of these involve the building of the infrastructures for the next NASA interferometric satellite[1] and for the National Virtual Observatory, a joint NASA, NSF undertaking.

Our objectives here are to involve JPL personnel and the Supercomputing facilities in early experiments with real applications to both test the suitability of IPG to our current problems and to set up a viable interaction with ARC IPG personnel with the aim to assist where possible in shaping the IPG capabilities towards these application requirements. Since the IPG is necessarily network based, our side objectives include stressing the networks between the two centers, JPL/ARC, as a means of assisting in their development as well.

Approach

During the course of these initial investigations, we have worked IPG concepts and resources for both the InSAR and NVO applications, using a simple, yet distributed computational model with plans to elaborate that model as we proceed. We have developed working relationships with ARC personnel in both the IPG and the NREN task teams. As part of this working relationship, we have installed the base IPG software - Globus - on the Castor Power Onyx and it is now considered to be an IPG asset.

Organization
We split the group up as follows:

Oversight	Bergman ,Curkendall, Tanner
Applications	Husman, Rosen, Siegel, Nigley
Data Movement	Becker, Miller, Kremenek ,Windoffer, Utley
Networking	Buchanan, Curkendall, de Luna, Freeman
Systems	Catherasoo, Hultquist, Utley, Windoffer

Chart 1

Additionally, we participated in an ARC/JPL demonstration of the jointly developed IsoWAN technology.

The following documents these accomplishments in more detail and outlines our recommendations for future activities.

NVO Prototype and the IPG - yourSky

The National Virtual Observatory (NVO) is being built by the astronomy community, with emphasis on many interoperable components, existing in a distributed high-performance, scalable environment. The clients themselves can be anywhere and do not need specialized or high performance capabilities. The dramatic growth in aperture and focal plane capabilities of institutionally managed observatories has resulted in an avalanche of data, both image and catalog. NVO is conceptualized to deal with these data by developing a layered, interoperable software architecture wedded to the Internet II with its national network of computational resources.

Architectural components of the NVO will need to include geographically distributed sites that are centers for Data Archive, Processing, and Visualization. We refer to the centers for Processing as Compute Nodes, which specialize in one or a set of perhaps computationally intense services. The Compute Node will be designed to help the user to organize his request, secure data from the archive services, secure the necessary compute cycles (from the Grid if necessary), perform the computations and return the results to the client node being manipulated by the end user.

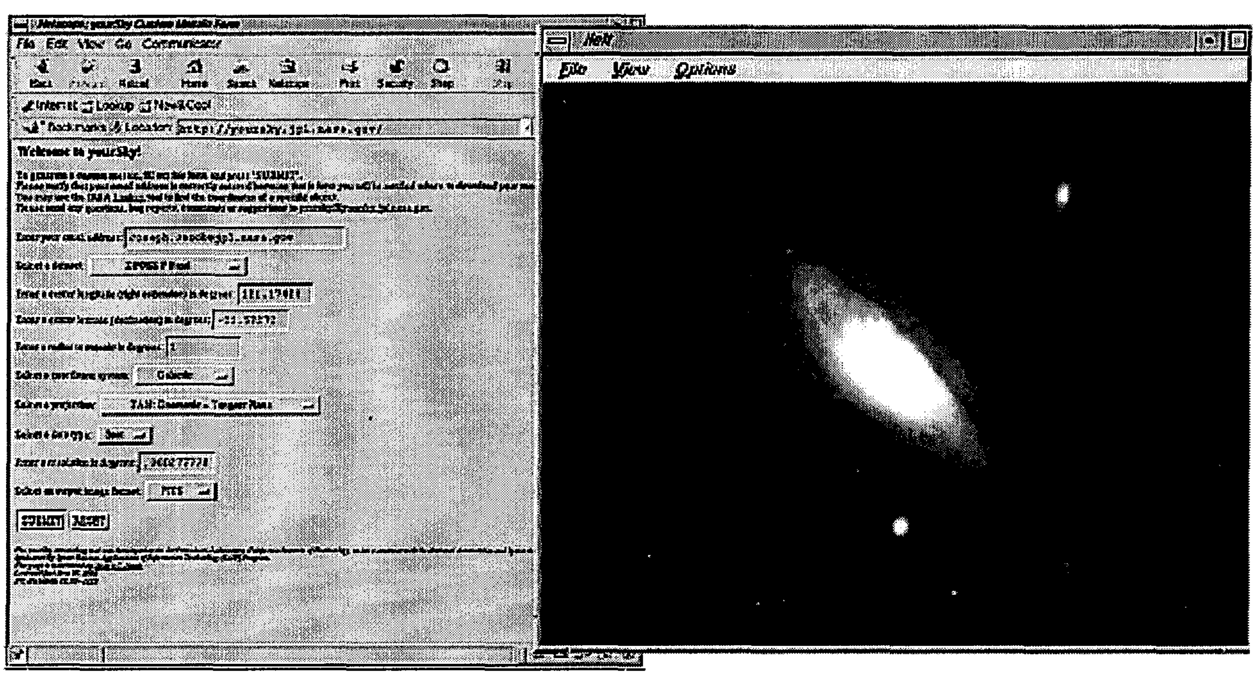


Figure 1. yourSky request and resulting image mosaic (DPOSS Andromeda).

As one instance of this Compute Node concept, we developed a custom image mosaicking engine called yourSky (http://yoursky.jpl.nasa.gov/). The interface to yourSky and a sample result (Andromeda) are illustrated in **Figure 1**. With yourSky, we have achieved a significant expansion of our image mosaicking capabilities with emphasis on *custom access, desktop accessibility* and *interoperability*. In this context, custom access refers to new technology that enables on-the-fly mosaicking to meet user-specified criteria for region of the sky to be mosaicked, data sets to be used, resolution, coordinate system, projection, data type and image format. Desktop accessibility was achieved by providing a web browser interface that permits the astronomer to submit a custom mosaic request and retrieve the result on a common desktop machine with only the ubiquitous web browser as a client. Interoperability refers to making yourSky interact, both as a data consumer and provider, with other software components of the NVO. As data consumer, yourSky is able to determine automatically which input images are required for a custom mosaic request. 10 TB of imagery from the Digitized Palomar Observatory Sky Survey (DPOSS) and the Two Micron All Sky Survey (2MASS) are accessible. Furthermore, yourSky retrieves the input images automatically from their respective tape archives. For DPOSS, this means data retrieval from the High Performance Storage System at Caltech's Center for Advanced Computing Research. The 2MASS image data are retrieved from the Storage Resource Broker (SRB) at the San Diego Supercomputer Center (SDSC). Our use of the SRB, described in more detail below, is an important milestone because it represents the first infusion of NASA's grid technologies into NVO activities.

Storage Resource Broker (SRB)

At http://www.npaci.edu/DICE/SRB, NPACI describes the SRB as "client-server middleware that provides a uniform interface for connecting to heterogeneous data resources over a network and accessing replicated data sets." The SRB can provide access to data stored in any of the following:

- a UNIX file system
- UNITREE, HPSS, and other archival storage systems, or
- Database objects managed by various DBMS including DB2 and Oracle.

SRB data may be accessed from remote clients in a number of ways. First, Unix-like utilities (e.g. ls, cp, chmod, etc.) are available for manipulating and querying data collections in the SRB space. Second, an API is available for sending requests from a client application and receiving a response from an SRB server. In addition, a Metadata Catalog (MCAT) server presents to clients a "logical view" of the data in the SRB. In simple terms, this means that the client uses an SRB file handle to access a file and does not need to know anything about the actual physical location of the data. Although data from the same collection may physically reside in different storage systems, the mechanics of how the data are accessed are transparent to client applications.

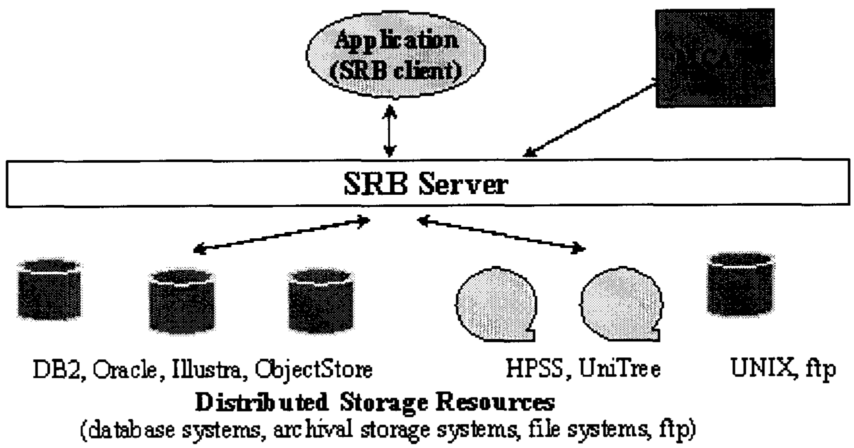


Figure 2

The SRB supports a number of authentication mechanisms including the Grid Security Infrastructure (GSI). In addition, capabilities are provided for auditing and monitoring use of the SRB. The architecture of the SRB is illustrated in **Figure 2**.

Accomplishments - NVO Application

Experiments

In FY01 the ESS program sponsored the first use of the IPG for NVO activities. As a first step toward integrating NASA's grid infrastructure with the NVO, we have extended the yourSky mosaicking server with automatic retrieval of 2MASS data from the SRB, described above. The end-to-end system makes about 5 TB of 2MASS imagery easily available for image mosaic construction by yourSky users.

In FY'01:

- Developed database that can be queried to determine which 2MASS images fall within a user-specified region of the sky.
- Developed a translator to convert from yourSky 2MASS file names to SRB file handle names.
- Demonstrated construction of 2MASS image mosaics with fully automated data retrieval from the SRB.
- In preparation for future work in this area, developed a working relationship with George Kremenek, a key SRB/IPG participant from SDSC.

Accomplishments - InSAR Application

Referencing the Organization **chart 1** of work breakdown, we see there are 5 subgroups, from JPL and ARC under the topics Oversight, Applications, Networking, Data movement, and Systems. For the most part each subgroup has been working largely on it's own. There is much expectation that the delayed but upcoming (Dec 01) IPG workshop will be the onset of a phase of co-ordination and integration. We discuss here progress made by specific sub-groups.

Systems-JPL: Pollux, a power onyx, with good networking and good storage has become a member of the IPG. The JPL local work was done by Cris Windofer

Application_JPL: As specified in the plan, Interferometic SAR processing of the ERS WINSAR data buy was the chosen application. Paul Rosen provided official permission for use to use this data. There are two repositories one at Caltech and one at Scripps. Craig Miller & Herb Siegel, have been successful at selecting and retrieving ERS data from Scripps. We have performed preliminary testing of data transfer performance. We have also ported the parallel version of ROI_PAC (current version minus one) to the Origins and performed preliminary performance and scaling tests.. In the table below you can see roughly how many nodes can be kept busy by network pipes of different bandwidths.

Since a productive distributed architecture must balance network speed and computational speed, we provide the following information on the times involved for transporting a single image pair of raw data to the target machine (~ 1300 mbytes), the computational time involved for focusing both images and processing through to interferograms, and the final transfer of the results (a similar 1,300 mbytes) to perhaps a third site.

Example Measured Network speeds: 3Mbytes/Sec - San Diego to JPL
12Mbytes/Sec - JPL to ARC

Computational performance: 16 nodes - 1 scene pair: 672 Seconds
32 nodes - 2 scene pairs: 753 Seconds

Time (seconds) to Transmit or Process Comparison Table (round trip transmit assumed)

	16 nodes	**32 nodes**
3Mbytes/s Scripps-JPL	867	1733
12Mbytes/s JPL-ARC	217	433
Processing Time	672	753

Chart 2

This table shows the importance of fast networks for the productive use of these architectures. The current Scripps JPL link is at less than OC-1 levels. Even if the target machine is a very modest 12 nodes, more time is spent networking than in computing. The rate from JPL at ARC really does work at the advertised OC-3 rates. Still, more than say 64 processors would not make a lot of sense. At OC-12, the processing could be fill even a 256 node machine and still be called 'balanced' and efficiently productive.

Accomplishments - IsoWAN

IsoWAN, a multi-center development led by JPL as part of the ISE program has been advanced as the methodology of choice to permit distributed, dynamic, collaborative teams to be formed within a very flexible, secure environment. [See paper referenced in the footnote]. As such, it is also being investigated by the IPG as a potential vehicle for securing IPG network transmissions when necessary. For example, if the IPG could be made secure, then the national complex of capability could be used for sensitive or even classified work such as is the case for SRTM processing.
In cooperation with this effort, the IPG investigation team participated in a Sept. 28th experiment between JPL and ARC. For this experiment, we implemented a cross platform MPI protocol and securely controlled tandem copies of JPL's Digital Light Table software for collaborative viewing of a SAR dataset, the JERS dual season Amazon mosaic. The demonstration was **very** successful and it is

expected that the IsoWAN will be incorporated in ARC's high performance networking alternatives and perhaps incorporated into the mainstream IPG software.

Significance

We look forward to employing the IPG to accomplish our goals of distributed high performance operation on terascale and larger datasets for both future InSAR missions and for the National Virtual Observatory. Achieving these goals for the first of these, InSAR, will enable the cost-effective deployment of a long term SAR satellite without incurring the cost of a dedicated, 'product oriented' SAR processing facility processing each and every scene and possible combinations thereof. These are timely topics as JPL's ESSP proposal for an Echo InSAR satellite has advanced to the status of a second round proposal. We envision a two tier Grid computing system to be appropriate for this mission. For traditional single issue InSAR processing, the ROI system can be distributed freely to scientists who request it. They would still need access to the data and we propose that instantiating these data worldwide under the SRB would be a system of choice. The second tier would be to enable a new class of large scale InSAR processing so that studies involving not individual interferograms but mosaics of interferograms ~1000km^2 . Then whole regions such as Southern California seismicity or large Antarctic ice flows could be monitored. Qualified scientists from access points perhaps around the world could log on to the Grid Computing Infrastructure, find the data they want, bring that data to capable computational facilities and have the final results returned to them. This two tier approach will mean that InSAR can realize its promise as a highly sensitive geodetic instrument that can be employed world wide to study plate tectonics and hazards (seismic, volcanic) monitoring.

Similar dataflow and computational schemes for the NVO will usher in a new era of data access and information discovery using the vast, world wide data resources that will comprise the NVO. It is clear that the NSF will be the center of gravity for the NVO. The standards being proposed and adapted for the NSF work are deeply embedded in the architecture and the specifications of the Grid. If, for example, our Round III ESS CAN efforts are to be fully relevant they must too be implemented to these standards. We are indeed fortunate that ARC's IPG deploys a friendly and supportive Grid development infrastructure to help with these issues.

Status/Plans

In FY02, we will continue to expand our use of IPG for both InSAR processing and for NVO activities. in two areas: (1) SRB data access, and (2) use of IPG computational resources for truly distributed, high performance computing.

NVO Plans

For 2MASS data access, we will continue to work with Kremenek (SDSC) to more tightly integrate SRB data access into the core JPL parallel mosaicking code that is invoked by yourSky. We currently access the SRB using the Unix-like client programs: Sinit to connect to the SRB, Scd to change the working directory to the 2MASS SRB directory, and Sget to retrieve a 2MASS image by name. Perl scripts are executed to do the data retrieval, the 2MASS images are staged at local disks, and then the parallel mosaicking code is invoked on the local data. In FY02, we would like to achieve tighter SRB integration by using the C API to read the 2MASS FITS format header and data blocks directly from the SRB, without requiring the need to first stage the data locally.

In FY02, we will also begin using the IPG computational resources for construction of custom image mosaics. When a yourSky custom image request is placed, the IPG will be used to find idle processors to handle the request at any of the participating NASA centers. The end of year deliverable will be to initiate a yourSky request from a random client on the internet, have the yourSky server act as an IPG client, dishing off the request to the available IPG processors, have the mosaicking code, running somewhere on the grid, access the 2MASS data from SRB and construct the mosaic, and return the resulting mosaic back to the yourSky server for retrieval by the end user. This end-to-end processing pipeline will demonstrate full integration of IPG services with yourSky.

InSAR Plans

These center on the perfection of the basic operations in the Grid context: access the data; transport and processes it remotely (target ARC) through to interferograms completion and then return to JPL for the more interactive and visual processing completion through to geodetic information. This will include the synchronization of the parallel ROI_PAC to the current single threaded software. We propose also to run under IsoWAN and test performance there, since using a secure transmission schema could open up sensitive data such as SRTM to distributed processing.

Finally, we will work with Networking-JPL/ARC sub groups. Experiments leading to agreed IPG network performance goals to support applications, with "Big data", will be published.

Point of Contact

Dave Curkendall
Jet Propulsion Laboratory
dwc@jpl.nasa.gov
818.354.2406

Source:
[1] Current plans propose an Echo satellite to be flown in the 2005-6 time period.

Previous Report List Next

Page last modified on 12/26/01

HTMT Phase 3 and Gilgamesh Transition

FY2001 Annual Report

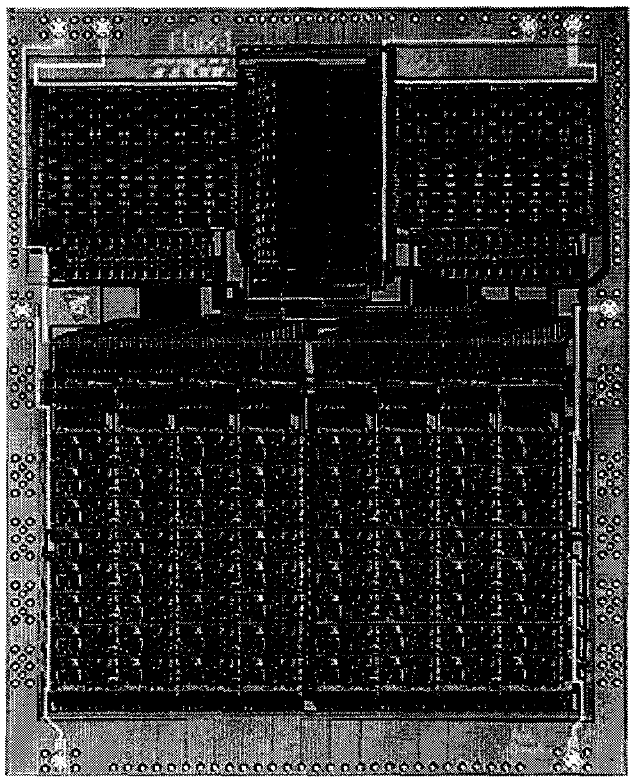


Figure 1. Flux-1 chip
(click image for enlargement)

Objective

The objective of the Gilgamesh task in FY'01 was to begin the effort of developing a Processor-In-Memory architecture for potential use in supercomputing systems, both for ground-based processing and in-flight applications.

Additionally, some continued oversight of the HTMT Phase 3 effort was provided, with the objective of completing the FLUX-1 demonstration of superconducting electronics and gracefully winding down the HTMT system software efforts.

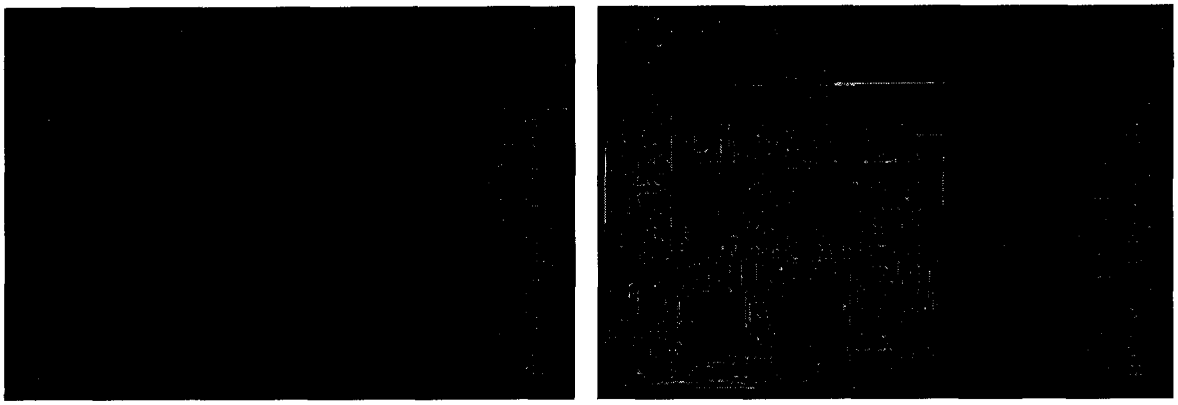


Figure 2. Gilgamesh FPGA board schematics
(click images for enlargement)

Approach

Gilgamesh is primarily a JPL/Caltech effort. Collateral efforts through Caltech involve Argonne National Laboratory and the University of Delaware in investigating software architectures for Gilgamesh-like architectures. Although currently unfunded, there is an ongoing collaboration with the University of Notre Dame.

Processor-in-memory (PIM) provides a promising approach to the growing gap between memory bandwidth and processor speed. Where the traditional separation of processor and memory to separate chips limits single memory accesses to 32- or 64-bits, PIM allows access to a full row of 2048 bits. The technical challenges are to develop a hardware architecture that can efficiently process data items in parallel while minimizing the physical area on-chip devoted to processor logic, and to develop a software approach that can make efficient use of data fetched from memory.

The Gilgamesh technical approach is to develop an FPGA-based prototype of the PIM architecture and use this board to evaluate architectural and hardware design choices, programming models, and the performance and resource requirements of simple application benchmarks. The majority of effort in FY'01 was focused on hardware efforts to design the prototype board, in developing a software implementation model to allow rapid turnaround from hardware design change to evaluation of simple application benchmarks, and in developing and modeling early architectural concepts.

As reported last year, HTMT is a multi-institutional effort to develop the technologies and architecture for a next-generation supercomputer. Funding constraints have limited this year's effort to providing guidance for the completion of the FLUX-1 demonstration effort. This demonstration is a collaborative effort between the State University of New York at Stony Brook, TRW, and JPL. Mikhail Dorojevets at SUNY is the processor architect, Paul Bunyk (who moved from SUNY to TRW this year) is the principal hardware designer, and TRW provides the fabrication facilities. At JPL, Arnold Silver and Alan Kleinsasser have facilitated an initially prickly relationship into one of close cooperation and high productivity.

Accomplishments

The design of the initial Gilgamesh prototype board has been completed and a review was held in September. This board will provide the basis for the initial architecture experiments.

A software development scheme has been devised to support rapid, interactive development and testing of Gilgamesh software. When coupled with the malleability of the FPGA hardware platform, this will allow very rapid progression from architectural concept through a reworking of the associated hardware logic to demonstration in simple applications.

Models of the initial Gilgamesh architecture have been developed; so far, these models indicate the criticality of processing as much data from a single memory access as possible, and for as long as possible. A 1:1 balance between processing time and memory access time is critical to achieving optimal utilization of PIM resources.

For HTMT efforts: the FLUX-1 chip has been fabricated, following several rounds of fabrication and testing of component logic designs. Testing has yet to be completed, but there is reason to believe that the chip will achieve a clock speed in the vicinity of 20 GHz. Loopback tests have shown that chip-to-chip communications speeds are likely to be as high or higher than can be achieved on-chip: 27 GHz has so far been observed in the laboratory, and the limiting factor has been the driver logic. With additional development and upgraded fabrication equipment--FLUX-1 has a feature size of 1.75 microns versus the CMOS state-of-the-art .18 micron feature size, and a 100x improvement in both logic density and speed may be achievable.

Significance

While there have been several other processor-in-memory efforts, most notably the Berkeley IRAM project and the IBM/Notre Dame efforts, Gilgamesh is the first attempt at developing a PIM architecture through a RAD approach. While previous efforts have resulted in prototype chips, most notably the IBM Execube developed by Peter Kogge, the limitations of current logic densities have limited PIMs to demonstration chips. PIM is an architecture for the future which may solve the problem of increasing logic density per chip, but relatively fixed pin counts. The Gilgamesh RAD approach may make it possible to realize a practical PIM architecture sooner.

The HTMT FLUX-1 demonstration chip is the first superconducting electronics part to achieve moderate complexity and clearly shows the potential for 100 GHz processor speeds. This would enable usable petaFLOPS when integrated into the HTMT architecture.

Status/Plans

With the design of the prototype FPGA board complete and delivery imminent, we expect development and demonstration of a prototype hardware architecture during FY'02. The initial processor logic design is nearly complete, with completed portions having passed simulation experiments.

The instruction set architecture, including both a "parcel" instruction set for communications and a processor instruction set are planned for FY'02. Through experimentation, these will be refined in the following years.

We have begun development of the high-level software concepts for efficient utilization of the Gilgamesh PIM. We expect to complete a preliminary software architecture definition in FY'02.

HTMT efforts under Gilgamesh have drawn to a close, but efforts will continue under renewed NSA funding. A FLUX-2 demonstration, showing multi-chip operation, is expected to commence early in FY'02.

Points of Contact

Loring Craymer
Jet Propulsion Laboratory
Loring.G.Craymer@jpl.nasa.gov
Phone: (818) 354-8355

Larry Bergman
Jet Propulsion Laboratory
bergman@kronos.jpl.nasa.gov
Phone: (818) 393-5314

Thomas Sterling
Jet Propulsion Laboratory/California Institute of Technology
tron@cacr.caltech.edu
Phone: (626) 395-3901

Page last modified on 12/26/01