Distributed High-Performance Computation for Remote Sensing

K.A. Hawick, H.A. James
Department of Computer Science, University of Adelaide, SA 5005, Australia
Tel +61 08 8303 4519, Fax +61 08 8303 4366,
Email khawick@cs.adelaide.edu.au, heath@cs.adelaide.edu.au

1 August 1997

Abstract

We describe distributed and parallel algorithms for processing remotely sensed data such as geostationary satellite imagery. We have built a distributed data repository based around the client-server computing model across wide-area ATM networks, with embedded parallel and high-performance processing modules. We focus on algorithms for classification, geo-rectification, correlation and histogram analysis of the data. We consider characteristics of image data collected from the Japanese GMS5 geostationary meteorological satellite, and some analysis techniques we have applied to it. As well as providing a browsing interface to our data collection, our system provides processing and analysis services on-demand. We are developing our system to carry out processing and data reduction services at-a-distance, enabling remote users with limited bandwidth, access to our system, to obtain useful derived data products at the resolution they require.

Our target hardware consists of a heterogeneous collection of distributed workstations, multi-processor servers and massively parallel computers at locations throughout Australia. These platforms are connected by ATM-based LANs and also through ATM switches across long distance WANs such as Telstra's Experimental Broadband Network, connecting Adelaide, Melbourne and Canberra. Our particular interest in constructing remote data access and processing services is the potential to utilise such resources as are available to a given user, yielding the best performance compromise of data processing and data delivery. To this end, we are building a set of resource scheduling and management utilities that will integrate the processing modules we describe. We have considered a number of software frameworks for building our integrated system and are focusing on a distributed object model using WWW protocols and the Java language.

Keywords: Remote-sensing, ATM, Classification, Client-Server, Satellite Imagery

1 Introduction

Integrating parallel and distributed computer programs into a framework that can be easily used by applied scientists is a challenging problem. Such a framework has to enable simplified access to both computationally complex operations and high-performance technologies, as well as providing a means for defining the appropriate data sets for the operation request.

Operations typically requiring such a system are computationally intensive and require fast access to very large large datasets. Rainfall prediction is one example of an application where the use of high performance technologies such as parallel and distributed computation is desirable. Near-time development of such predictions is of particular interest to land managers requiring accurate harvesting and planting decision support, as well as resource and financial managers interested in likely crop yields and futures predictions.

Our Distributed High-Performance Computing project is part of the Research Data Networks Cooperative Research Centre and is fortunate to have access to a number of distributed and parallel computing resources both in Adelaide and in Canberra. Distributed remote resources are interconnected using a high speed network and running remote interactive jobs is entirely feasible. We believe these resources are representative of the heterogeneous computing environment that is available to users in Australia and our DISCWorld[9] project is building a software environment for integrating together applications clients with high-performance computational service components to allow the best or fastest computing resource to be used.

We describe an application component for a demonstrator of the computational requirements of a DHPC system. We envision a system whereby environmental scientists or value-adders will be able to utilise a toolset as part of a DISCWorld system to prepare derived-data products. End-users will be able to access these products in pre-prepared format or even request them interactively from a low-bandwidth World Wide Web browser platform such as a personal computer. Web protocols will be used to specify a complex service request and the system will respond by scheduling high performance computing (HPC) and data storage resources to prepare the desired product prior to delivery of the condensed result in a form suitable for decision support. HPC resources will be interconnected with high performance networks, whereas the end user need not be. In summary the end-user will be able to invoke actions at a distance on data at a distance.

We describe our ATM-connected network system in section 2 and the importance of bandwidth and latency in section 3. We discuss image processing applications processes in section 4 and some of the parallel computing issues surrounding them in sections 5 and 6.

2 ATM and the EBN

Asynchronous Transfer Mode (ATM) [8] is a collection of communications protocols for supporting integrated data and voice networks. ATM was developed as a standard for wide-area broadband networking but also finds use as a scalable local area networking technology. ATM is a best effort delivery system – sometimes known as bandwidth-on-demand, whereby users can request and receive bandwidth dynamically rather than at a fixed predetermined (and paid for) rate. ATM guarantees the cells transmitted in a sequence will be received in the same order. ATM technology provides cell-switching and multiplexing and combines the advantages of packet switching, such as flexibility and efficiency of intermittent traffic, with those of circuit switching, such as constant transmission delay and guaranteed capacity.

Although a number of wide area broadband networks have been built in the USA [11, 19], it is unusual to have a network that is fully integrated over very long distances rather than local area use of ATM technology. Telstra have built the Experimental

Broadband Network (EBN) [4] to provide the foundation for *Australian* broadband application development. Major objectives of this are to provide a core network for service providers and customers to collaborate in the development and trial of new broadband applications, and to allow Telstra and developers to gain operational experience with public ATM-based broadband services.

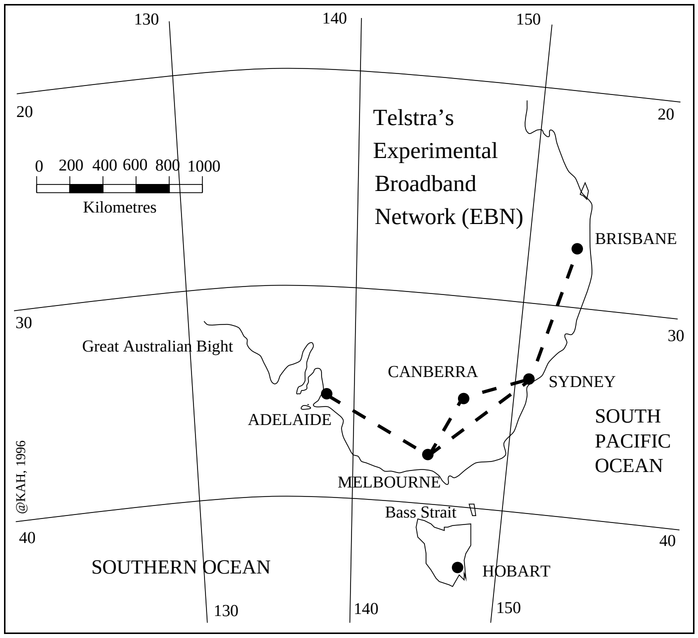


Figure 1: Telstra's Experimental Broadband Network (EBN)

The current layout of the EBN is shown in figure 1 and some of its latency and performance aspects are discussed below.

3 Latency and Bandwidth

We have experimented extensively with our resources at Adelaide and Canberra in assessing the capabilities of a distributed high performance computing system operating across long distances. It is worth considering the fundamental limitations involved in these very long distance networks. Telstra's EBN is shown in figure 1. Currently it connects Adelaide, Melbourne, Sydney, Canberra and Brisbane.

Although we employ OC-3c (155Mbps) multi-mode fibre for local area networking we are restricted to an E-3 (34Mbps) interface card to connect Adelaide to Melbourne and hence to Canberra.

The line-of-sight distances involved in parts of the network are: Adelaide–Melbourne 660 km; Melbourne–Canberra 467 km; Melbourne–Sydney 710 km; and Sydney–Brisbane 732 km. Consequently the effective network distances between Adelaide and the other cities are shown in table 1.

The light-speed-limited latencies shown in table 1 are calculated on the basis of the vacuo light-speed ($2.9978 \times 10^5 \mathrm{km} s^{-1}$). It should therefore be noted that this is a fundamental physics limitation and does not take into consideration implementation details, the most important of which are that the EBN does not necessarily use fibre over its entire length, the actual route used has distances that are almost certainly

EBN City	Network Distance from Adelaide (km)	Light-speed-Limited Latency (ms)
Melbourne	660	2.2
Canberra	$660 + 467 = 1127$	3.8
Sydney	$660 + 710 = 1370$	4.6
Brisbane	$660 + 732 = 2102$	7.0

Table 1: Inter-city distances (from Adelaide) and Light-speed-limited latencies for the Experimental Broadband Network.

longer than the 'city route-map' ones quoted here, and that electron carried signals propagate more slowly than light-speed. The important point is that over this length of network some significant latency is unavoidable and applications developed to run over such distances must be developed with this in mind.

Performance measurements have been made using widely-available network benchmarking tools. The results of these test are shown in table 2. By varying the packet sizes sent it is possible to derive crude latency and bandwidth measurements. It should be emphasised these measurements are *approximations* of what is achievable and are only for comparison with the latency limits in table 1.

Ping Packet Size (Bytes)	Mean Time Canberra via EBN (ms)	Mean Time Syracuse USA via Internet (ms)	Mean Time local machine via ethernet (ms)	Mean Time local machine via ATM switch (ms)
64	15	340	0	0
1008	16	367	3	0
2008	17	387	6	1
4008	18	405	9	1
6008	20	428	14	1
8008	22	441	18	2

Table 2: Approximate Performance Measurements using Ping.

The times in table 2 are averaged over 30 pings and represent a round-trip time. Measurements are all to a precision of 1 ms except for those for Syracuse which had a significant packet loss and variations that suggest an accuracy of ± 20 ms is appropriate.

The *ping* measured latency between Adelaide and Canberra appears to be approximately 15 ms. This is to be compared with the theoretical limit for a round-trip of 7.6 ms. The EBN appears to provide close to the best reasonably achievable latency.

Also of interest is the bandwidth that can be achieved. The actual bandwidth achieved by a given application will vary depending upon the protocols and buffering layers and other traffic on the network but these *ping* measurements suggest an approximate value of $2 \times 8\text{kB}/(22 - 15 = 7\text{ms}) = 2900kB/s \simeq 22.7\text{Mbps}$. This represents approximately 84% of the 27Mbps of bandwidth available to us, on what is an operational network.

A typical achievable bandwidth between local machines on the operational 155Mbps fibre network is 110.3 Mbps compared with a typical figure on local 10Mbps ethernet of 6.586Mbps. Both these figures are representative of what was a busy network with other user traffic on them. We believe that the maximum theoretical bandwidth running Classical IP over ATM networks is 135.0 Mbps [16].

4 Image Classification

We describe characteristics of image data collected from the Japanese GMS5 geostationary meteorological satellite, and some analysis techniques that can be applied to it. We also discuss some of the information products that may be derived from the data and how these can provide input for other applications for various value-adders and end-users.

The GMS5 satellite simultaneously produces a set of four images, each of a different wavelength, by the use of the onboard Visual and Infra-Red Spin Scan Radiometer (VISSR) [20]. The different channels of data represent different parts of the spectrum, and along with the repository that stores the data are described in [13]. Examples of the images produced are shown in figure 2.

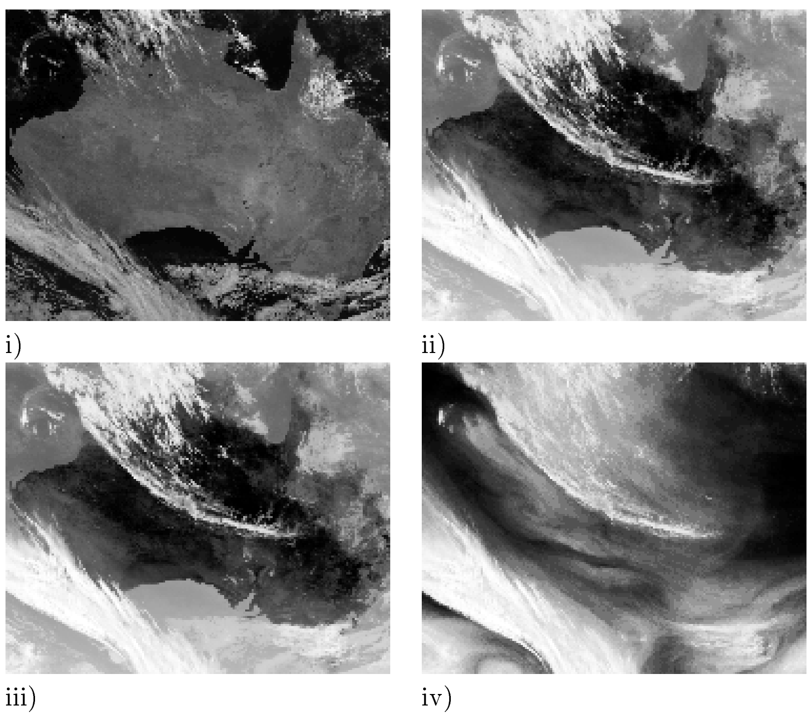
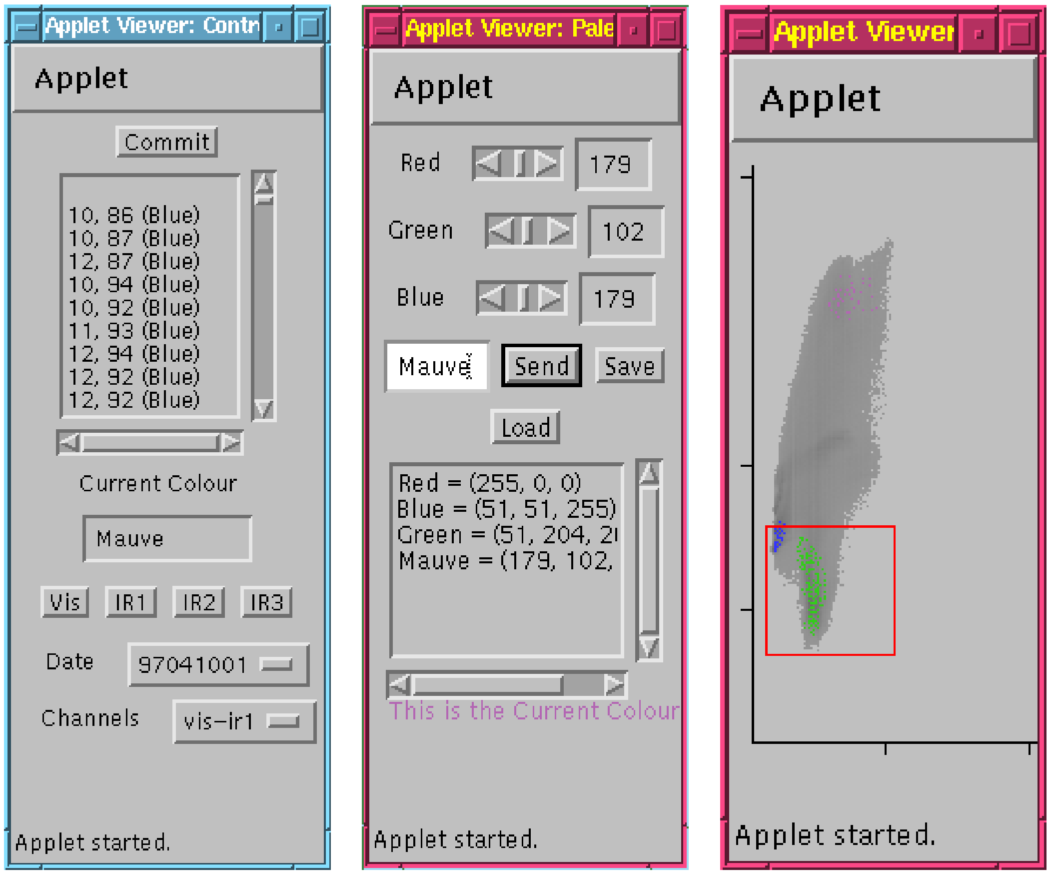


Figure 2: GMS-5 Data: i) Visible Spectra, ii) Thermal IR1 Spectra, iii) Thermal IR2 Spectra and iv) Water Vapour IR Spectra

It is possible to create correlation plots between pixel values in different channels of data. These plots show the frequency of a pixel value set occurring in each channel. For example, as the wavelengths of the IR1 and IR2 bands are very close, the correlation between the pixel values represented in the images is very high. Conversely, the correlation between the Visual and other channels is not high, and is somewhat localised, as shown in figure 3(iii).

We have build a Java [1] application to allow a user to classify and colour regions occurring in multiple channels of an image set. This application has been designed to be a front-end for a larger system, which will automatically classify regions of interest in multiple data sets.



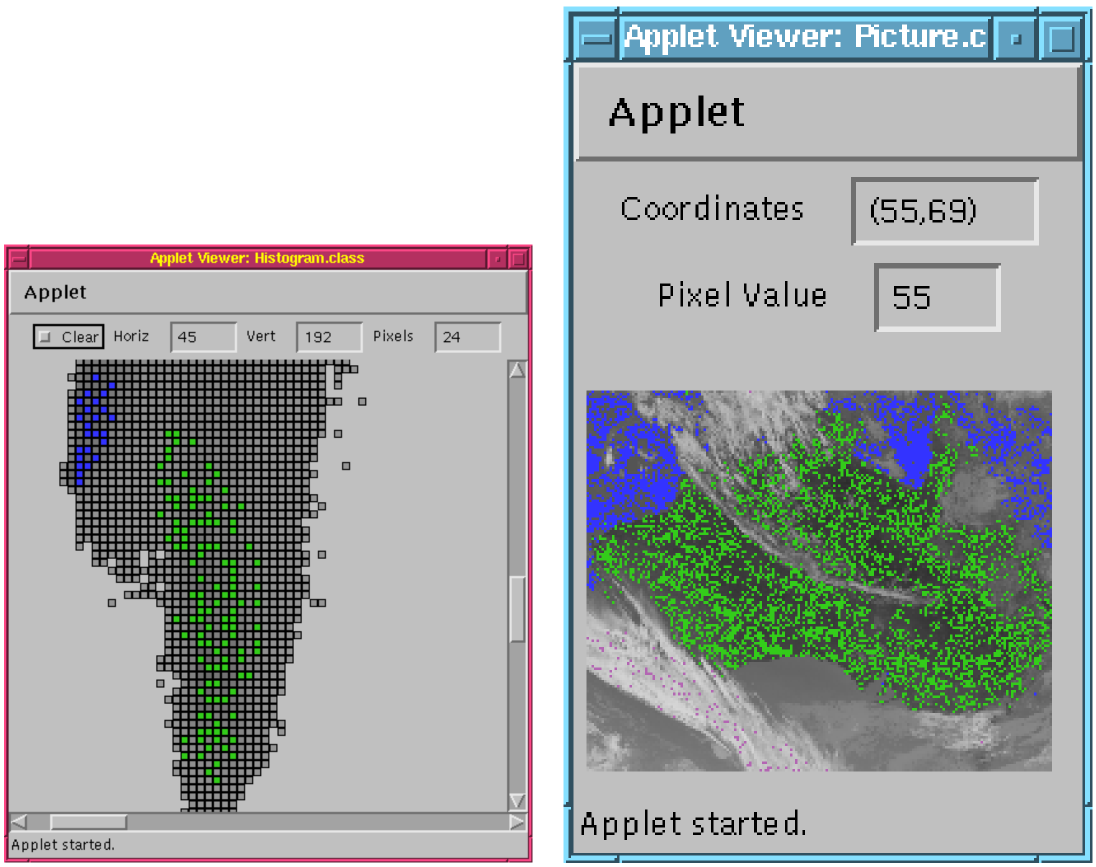
i) ii) iii)

Figure 3: Java Image Classifier: i) Control Panel, ii) Colour Palette and iii) Main Histogram

By use of the control panel, as shown in figure 3(i), the user can select the data set with which to work, the images within that set to use for the correlation graph, and the image on which to superimpose the pixel colouring. The application loads a set of images, displaying the image selected in the control panel, and produces correlation plots of the selected channels' pixel values. These are displayed in a histogram such that the frequency of the pixel set occurrence is indicated by the darkness of the boxes, as shown in figures 3(iii) and 4(iv).

The colour palette, shown in figure 3(ii), allows the user to select a colour that will be used to highlight the region of interest. The user is presented with a zoomed histogram, figure 4(iv), in which they can select individual pixels. The action of selecting pixels in the zoomed histogram causes all pixels corresponding to that location to be changed to the current colour in the image viewer (figure 4(v)).

A list of the pixel sets that have been modified is stored; this list is recorded in a file,



iv) v)

Figure 4: Java Image Classifier: iv) Magnified Histogram, v) Image Previewer

which may be used as input to a system that will automatically colour the pixels across different image sets. This automatic colouring will enable us to produce animated sequences of images, thus allowing us to visualise weather patterns as they progress in time and other land classifications.

It is possible, with some degree of accuracy, to derive the correlation of sub-images. For example, high-altitude cloud and ground have quite different correlation graphs. If the correlation graph of high-altitude cloud is known in advance, and is selected in the correlation graph of the whole image, this should result in the selection of all the high-altitude cloud for the image. If this is done for all the major components of the image, ie sea, high- and low-altitude cloud, and land, then it should be possible to almost fully classify and colour all regions of the image.

If it is found that at the starting stages of a cyclone, there is a certain correlation graph, then image sets may be automatically scanned to contain that graph. This may be useful information for automated forecasting.

Figure 5 shows some of the interesting features from a simple land/sea/cloud classification using only two channels - visual reflected signal and thermal infrared emitted signal. In this case a very simple scheme has been employed to identify regions in the correlation plot of the two channels, grouping simple polygons of pixel values as “features” and colouring them either brown/green for land; blue for sea or white/grey for clouds. A specular reflection effect in the visual channel arising from an area of sea shows up as an anomalous pseudo-land feature in the classified colour-composite image.

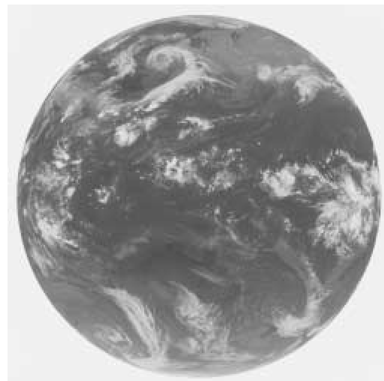
4.1 Image Pre-Processing and Scanline Corrections

The GMS satellite imagery we receive is generated as a set of scanlines. Occasionally a scanline dropout occurs from transmission errors for example and a black or empty image line appears. In the case of single dropout lines it is possible to interpolate lines above and below to acceptable accuracy for some product purposes. For generating video sequences of the imagery however, a more robust approach is required to correct multiple dropout line errors. A useful technique is to Fourier Filter the image, and apply a wedge filter to remove artifacts in the spectrum that correspond to the dropped lines. This can be used to generate approximations to the dropped lines based on neighbouring values. We are experimenting with various parallel algorithms for applying Fast Fourier Transforms to what are large image sizes - specifically 2291×2291. These sizes are also inconveniently non-powers of two which requires padding or truncating the images. We are also investigating non-radix-two algorithms for this purpose.

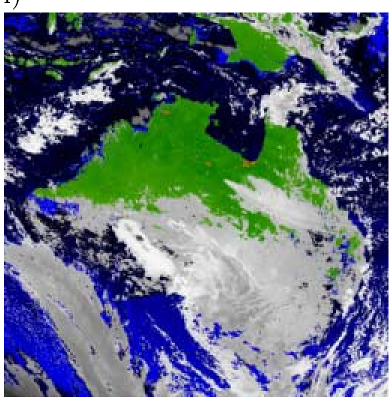
4.2 Image Remapping and Coordinate Transforms

The imagery in our repository can be transformed from what is essentially raw pixel coordinates to a mapping onto the Earth’s surface using the known geometry, orbit and attitude of the satellite. We are constructing a data parallel computational module that performs this transformation on image data. This is a computationally intensive task, requiring a significant number of trigonometrical calculations to be carried out for each pixel transformed. We are experimenting with this as both code for the CM5 as well as for a loosely coupled farm of Alpha workstations. Two approaches are useful. The satellite image can be rectified pixel by pixel to map it onto longitude and latitude. This requires a series of trigonometrical and matrix operations to be carried out for each pixel. A computationally cheaper approach is

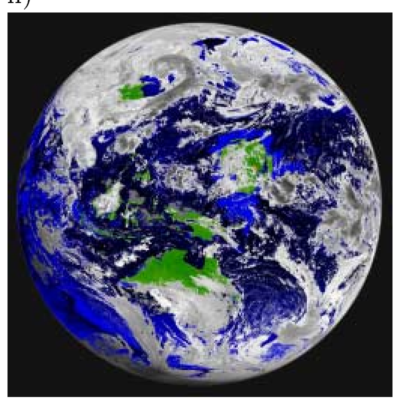
i)



ii)



iii)



iv)

Figure 5: GMS-5 Data: i) Visible Spectra, ii) Thermal IR1 Spectra, iii) Thermal IR2 Classified Australia and iv) Classified World

to warp an image of the desired earth map onto the image as it would be seen by the satellite. This can be carried out by applying a mapping to the vector based data in a land sea mask for example and requires considerably fewer operations than a mapping of the entire GMS5 2291×2291 pixel image. Figure 6 shows a typical mapping of the land sea mask around Australia as it would be seen by the GMS5 satellite.

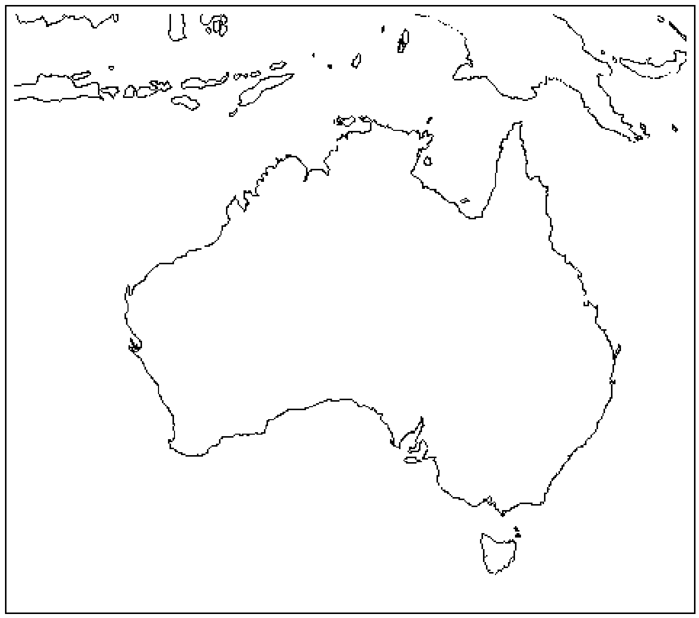


Figure 6: Land Sea Mask over typical region as seen by GMS5 Satellite

4.3 MPEG Movie Sequences On-Demand

MPEG uses reference frames, and predicted frames, and the differences between them. Motion is predicted between frames in the temporal direction, and difference cosine transforms (DCT) are used to organise the redundancy in the spatial directions. There are three types of frames: I, B and P frames. I-frames are "intra" frames. They are a frame coded as a single still image, with no history and are used for "setting the stage". From these P-frames are obtained, which are the predicted frames. P-frames are predicted from the most recently reconstructed I or P-frame. Depending upon the match between the current P-frame and the last I or P-frame, it can be coded just like an I-frame, or will come with a vector and DCT coefficients for a close match.

Finally, B-frames are bi-directional frames and are constructed by taking the two closest I or P-frames (one in the past and one in the future) and use one of three methods to find matching blocks in the frames. If none of the methods works, then it is legal to encode the frame as an I frame. A typical sequence that we employ for encoding the satellite imagery is: IBBPBBPBBPBBPBBP. The software infrastructure for constructing MPEG movies on-demand is described in more detail in [3] .

5 Parallel Components

High performance processing of the image manipulation algorithms is achieved through embedded parallelism. We employ HPF [14], CC++ [21] and MPI [7] parallel modules for manipulating the large fields of image data [15].

Part of our research involves the designing and building of a parallel, distributed image processing library. This library consists of a number of modules, each of which will be able to utilise the supercomputing facilities that we have access to, namely a Thinking Machines CM-5, Silicon Graphics Power Challenge, and two AlphaStation farms (in Adelaide and Canberra).

6 Distributed Computing and On-Line Processing

Parallel computational modules in our system are integrated together using a client-server model. Of particular interest to us is chaining processing requests together so that all data intensive calculations can be performed remotely. We have built a Java front end to individual processing modules which are each managed by a Java server. We are investigating ways to allow server to communicate together peer-to-peer to satisfy a complex user processing request.

For example, a catalog server might respond to a user request to specify a sequence of images to extract from the repository. The metadata specifying this sequence might be passed to the repository storage manger server which would feed the images to a temporary staging area. The user might have specified a classification process to be applied to the entire sequence using the classification applet described above. The classification matrix can then be applied to the entire sequence of images specified by a classification server engine. Finally, a the resulting artificially coloured images might be merged together into a movie sequence by a different server. We are investigating protocols for servers to be managed in a hierarchy, with user requests being serviced through an interface server.

7 Discussion and Conclusions

We have identified a number of applications of parallel and distributed computing for supporting on-line processing of remotely sensed data. Image processing and analysis is a particularly rich field for providing a rich set of transforms and analysis application components for integration in a distributed computing system. We have measured good speed-up performance for the parallel and distributed components of our system and conclude that the Java Remote Method Invocation technology is well-suited to building an infrastructural "glue" for building distributed, high-performance computational systems.

Acknowledgements

We thank K.P.Bryceson, K.J.Maciunas and F.A.Vaughan for valuable discussions on design of the data storage and delivery system and K.J.Hutchens for coding part of the image classification system.

We acknowledge support from the Research Data Networks Cooperative Research Centre(CRC) and the Advanced Computational Systems CRC of the Australian

Commonwealth Government.

References

[1] “The Java Language Specification”, James Gosling, Bill Joy, Guy Steele, Pub. Addison Wesley, The Java Series, 1996, ISBN 0-201-63451-1.

[2] “Fundamentals of Remote Sensing and Airphoto Interpretation”, T.E.Avery and G.L.Berlin, 5th Edition, Pub. Macmillan, 1985, ISBN 0-02-305035-7.

[3] “Movie Sequences of Archived Imagery on Distributed, High-Performance Computing Systems”, P.D.Coddington, K.A.Hawick, H.A.James, F.A.Vaughan, Technical Note DHPC-012, August 1997.

[4] “Telstra’s Experimental Broadband Network”, D.Kirkham, Telecommunications Journal of Australia, Vol 45, No 2, 1995.

[5] “Berkeley MPEG-1 Video Encoder: User’s Guide”, Kevin L. Gong, Computer Science Department, Berkeley, March 1994.

[6] “Digital Image Processing”, Rafael C. Gonzalez and Richard E. Woods, Pub. Addison-Wesley, 1992, ISBN 0-201-50803-6.

[7] “Using MPI: Portable Parallel Programming with the Message-Passing Interface”, W. Gropp, E. Lusk, A. Sjkellum, Pub. MIT Press, ISBN 0-262-57104-8, 1994.

[8] “ATM Networks - Concepts, Protocols, Applications”, R.Handel, M.N.Huber, S.Schroder, Pub. Addison-Wesley, 1994, ISBN 0-201-42274-3.

[9] “DISCWorld: A Distributed Information Systems Control System for High Performance Computing Resources ” K.A.Hawick, K.J.Maciunas and F.A.Vaughan DHPC Technical Report DHPC-014, 1997. Department of Computer Science, University of Adelaide.

[10] “A Strategy and a Tool, Ferret, for Closely Integrated Visualization and Analysis”, S.Hankin, D.E.Harrison, J.Osborne, J.Davison, and K.O’Brien, http://www.pmel.noaa.gov/ferret/rostock_paper/paper.html, 1996.

[11] “Applications and Enabling Technology for NYNET Upstate Corridor”, S.Hariri and G.Fox, Northeast Parallel Architectures Center (NPAC), Technical Report SCCS-642, November, 1994.

[12] “Introduction to Remotely Sensed Data” B.A.Harrison and D.L.B.Jupp, CSIRO, Division of Water Resources, 1989, ISBN 0-643-04991-6.

[13] “Geographic Information Systems Applications on an ATM-Based Distributed High Performance Computing System”, K.A.Hawick, H.A.James, K.J.Maciunas, F.A.Vaughan, A.L.Wendelborn, M.Buchhorn, M.Rezny, S.R.Taylor and M.D.Wilson, Proc. HPCN, Vienna, Austria, 1997. Also DHPC Technical Report DHPC-003.

[14] “High Performance Fortran Language Specification”, High Performance Fortran Forum (HPFF), *Scientific Programming*, vol.2 no.1, July 1993.

[15] “Stencil Methods on Distributed High Performance Computers”, H.A.James, C.J.Patten and K.A.Hawick, DHPC Working Note, May, 1997.

[16] "Analyzing ATM Adaptor Performance. The Real-World Meaning of Benchmarks", C.D.Krivda, Efficient Networks, 1996, http://www.efficient.com/dox/EM.ps

[17] "Remote Sensing and Image Interpretation", T.M.Lillesand and R.W.Kiefer, 3rd Edition, Pub. Wiley, 1994, ISBN 0-471-30575-8.

[18] "Applied Remote Sensing", C.P.Lo, Pub. Longman, 1986, ISBN 0-582-30132-7.

[19] "CASA Gigabit Network Testbed: Final Report", P.Messina and T.Mihaly-Pauna, Eds, CASA Technical Report 123, July, 1996.

[20] "The GMS User's Guide", Pub. Meteorological Satellite Center, 3-235 Nakakiyoto, Kiyose, Tokyo 204, Japan. Second Edition, 1989.

[21] "A Tutorial for CC++", P.A.G.Sivilotti and P.A.Carlin, Department of Computer Science, California Institute of Technology, Technical Report CS-TR-94-02, 1994.