High-Throughput Computing To Enhance Intervisibility Analysis

M.J.Mineter[1], S.Dowers[1], D.R.Caldwell[2], B.M.Gittings[1]

[1] Geography, School of GeoSciences, University of Edinburgh,
Drummond St, Edinburgh EH8 9XP, Scotland
Tel. +44 131 650 2662
Fax. +44 131 650 2524
{mjm, sd, bruce}@geo.ed.ac.uk

[2] US Army Research and Development Center,
Topographic Engineering Center, 7701 Telegraph Road,
Alexandria, VA, USA 22315-3864.
Tel. +1-703-428-6802 X2236.
Fax. +1-703-428-6425.
EMail. Douglas.R.Caldwell@erdc.usace.army.mil

Biography

Research fellow at the University of Edinburgh 1992 – 2001; from January 2003. At JRC-Ispra, Italy July 2001 – Dec. 2002, developing GIS related to sustainability analyses. Primary interests: applying high performance and Grid computing to GIS, environmental modelling, decision support and policy-making.

Introduction

The viability and accessibility of high performance computation is increasing as never before, and yet the application of these technologies in GIS remains rare. The expectations of those developing GIS applications are still largely determined by the capabilities of commercial software running analyses on single processors, albeit with benefits from continuing advances in processor power, powerful servers and web services offering both data and computation. As argued elsewhere (Dowers et al, 2000) components implemented on high performance architectures could now be routinely integrated with GISystems, enabled by the emerging standards from the OGC and also the Web service and Grid computing communities.

Within high-performance computation, a classification can be made into parallel processing (in which a single task is shared by multiple processors) and high-throughput computing (in which many related tasks are each run without synchronisation, each on its own processor.) Well-known examples of the latter include SETI@home (2002) and systems built upon Grid middleware such as APST (2002) for parameter sweeping.

As Kidner (2001) has written in the context of interpolation algorithms, the availability of high performance reopens scope for approaches that had previously been discarded as impractical. Intervisibility is such an example, for in analyses based on visibility it is routine to seek relatively complex algorithms to reduce computation times. We explore the alternative route, in applying high-throughput computing to intervisibility analyses, so achieving relative simplicity and opening possibilities for new applications using intervisibility data.

The emphases are on

The use of high-throughput computing to create a "Complete Intervisibility Database" (CID) comprising a bitmap termed a Masked Area Plot, for each post of a Digital Elevation Model, the bitmap identifying the visible regions of the DEM. (Uncompressed, this would be $(nRows*nCols)^2$ bits.)

The CID structure.

Integration of applications that access the CID with a GISystem. The integration in the current implementation is on the user's desktop processor.

High-throughput computing platform for Intervisibility

An initial demonstrator (Mineter et al., 2002) was built upon an in-house system, GANNET (Geographical ANalyses on NETworked computers) that distributes ArcInfo tasks to multiple UNIX processors. A 'task farm' approach was taken: one processor acts as coordinator of the task farm, and each processor requests a new task when it is ready to process it, so ensuring the necessary flexibility in response to changing loads on different processors. The coordination is achieved using PERL, remote shells, and shared directories. The tasks are run at low priority to use spare processing capacity, and hold all data on local disk space and then copy results to a final directory on a shared disk. A single task comprised the visibility analysis for 16 posts in a DEM, invoked in ArcInfo (installed on each processor) by use of AML scripts. For analysis of a DEM of size 366 by 466 posts a single task had a typical run time of between 2 and 5 minutes, depending on which processor was used: of the 13 available, 5 were 300MHz Sparcs, the remainder being older Digital processors. Over holiday periods, these were used with efficiencies between 93% and 98% so reducing the elapsed time to 43hours 26 minutes.

In a second phase GANNET was integrated with the Condor (2002) middleware, allowing use of 40 more powerful processors running Windows. Recent releases of Condor have enhanced its support for Windows, offering solutions to the issues concerning the invocation of remote shells on Windows, and also providing a capability for additional computation in other applications (for example, by supporting the parallel processing message-passing standard, MPI).

Complete Intervisibility Database

A Masked Area Plot for the post at column i and row j, MAP(i,j) contains a binary value for each post in the grid indicating whether that post is visible from the originally selected post at (i,j). The MAP therefore contains nCols x nRows binary values. A Complete Intervisibility Database (CID) is a collection of MAP(i,j) for each post in the original grid. There are therefore nCols x nRows MAPs in the CID.

The Complete Intervisibility Database is a collection of directories, containing files where each holds 16 MAPs. Each file holds a 16-bit integer grid, the k'th bit being part of the k'th MAP. The grid is converted to a Band Interleaved by Line (BIL) format, compressed using zip and copied to the appropriate directory in the Complete Intervisibility Database. The folder and file name is determined from the location of the 16 observers. By selecting the 16 observers in a 4x4 sub-grid of the original grid we can expect there to be significant correlation between the bits in a 16-bit word, and so improved compression. This database structure has the advantage of simple random access to any BIL file and expansion can be achieved using standard tools available on almost all computing platforms.

In a trial for an input DEM of 336 columns by 466 rows, the compression reduced the dataset size to 84Mbytes, ~3% of the uncompressed size (2.9Gb). In this case the CID comprises 9828 files, with 84 files in each of 117 directories (one directory per sub-grid row).

Applications built on the CID

Java applications were developed to interrogate and process the CID. These generate descriptive surface measures, such as cumulative visibility, and tactical decision aids, such as percent target visible and are described elsewhere (Caldwell et al., Geocomputation 2003 poster).

Visualisation of the results is within the ESRI ArcMap application, the Java running on the same processor. Extensions to ArcMap have been written using Visual Basic with the ArcObjects interfaces. The CID is accessed across the network as a shared disk. The VB code runs a shell that invokes the Java, which writes data to temporary files and then uses a named pipe to communicate completion of the shell to the VB. The data are then read from the temporary files and displayed in ArcMap.

Summary

The use of high-throughput computing makes tractable the run-times inherent in generating a "Complete Intervisibility Database" by enabling the concurrent execution of multiple ArcInfo commands on multiple processors. Readily available methods of compression have been demonstrated in reducing data volumes to easily manageable levels: for a test dataset of 366 by 466 posts both the Java applications and the CID can be held on one CD. Integration of the resulting CID with the ArcGis desktop was accomplished by coupling ArcMap to applications written in Java.

References

APST (2002) AppLeS Parameter Sweep Template http://grail.sdsc.edu/projects/apst/index.html accessed 28 Feb 2003

Condor (2002) The Condor Project Homepage http://www.cs.wisc.edu/condor/ accessed 28 Feb 2003

Dowers S., B.M.Gittings, M.J.Mineter, (2000), Towards a framework for high-performance geocomputation: handling vector-topology within a distributed service environment, Computers, Environment and Urban Systems, 24, 471-486

Kidner, D. (2001) DEM Interpolation – why settle for less? Geocomputation 2001, http://www.geocomputation.org/2001/papers/kidner.pdf accessed 28 Feb 2003

Mineter, M., Dowers, S., and Gittings, B. (2002) Software Infrastructure to Enable Parallel Spatial Data Handling: Final Technical Report. R&D 8707-EN-01 Contract N68171 00 M 5807. London: U.S. Army Research Office.

SETI (2001) Search for Extraterrestrial Intelligence http://setiathome.ssl.berkeley.edu/ accessed 7 Dec 2002