A Parallel PC-based Visible Human Slice WEB Server

Samuel Vetsch, Vincent Messerli, Oscar Figueiredo, Benoit Gennart, Roger D. Hersch

École Polytechnique Fédérale, Lausanne, Switzerland

{info@visiblehuman.epfl.ch}   {http://visiblehuman.epfl.ch}

Laurent Bovisi, Ronald Welz

WDS Technologies SA, Geneva, Switzerland

Luc Bidaut

Hôpital Cantonal Universitaire, LFMI, Geneva, Switzerland

Visualization of 3D tomographic images by slicing, i.e. by intersecting a 3D tomographic image with a plane having any desired position and orientation is a tool of choice both for learning and for diagnosis purposes.

In this project, a parallel Visible Human Slice Web server has been developed, which offers to any Web client the capability of interactively specifying the exact position and orientation of a desired slice (Figure 1) and of requesting and obtaining that slice from a 3D tomographic volume, made of either CT, MRI or cryosection images (digital color photographs of cross-sections). For interactive slice position and orientation, a miniature 3D version of the full image is used, as in [North96].

Figure 1. Selecting within a Java applet an image slice within a miniaturized 3D tomographic image

Accessing and extracting slices from the 3D Visible Human image [Ackerman95] requires considerable storage space, e.g. 13GBytes for the human male dataset, and high processing power. We decided therefore to build the Visible Human Slice Web server on top of 5 Bi-Pentium Pro 200MHz PC's. An additional Bi-Pentium II 333MHz PC acts as the Web server running Microsoft IIS 4.0. All the PC's are interconnected through a 100Mbits/s Fast Ethernet switch. A total of 60 2GB disks is equally distributed among the server's 5 Bi-Pentium Pro PC's.

For enabling its parallel storage and access, the Visible Human 3D volume is segmented into volumic extents of size 32x32x17 RGB voxels, i.e. 51KBytes, which are striped over the 60 disks residing on the 5 server PC's. In order to extract an image slice from the 3D image, the extents intersecting the slice are read and the slice parts contained in these volumic extents are extracted and projected onto the display space (Figure 2).

volumic extent

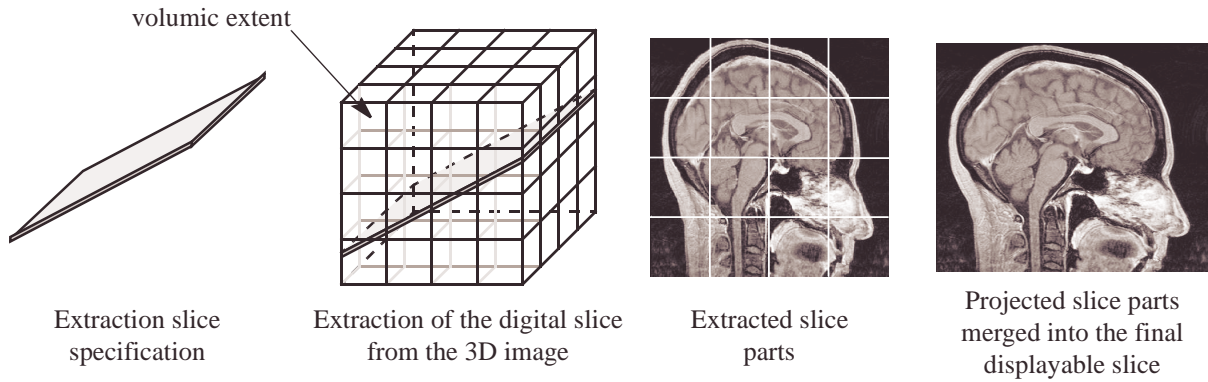| Extraction slice specification | Extraction of the digital slice from the 3D image | Extracted slice parts | Projected slice parts merged into the final displayable slice |

Figure 2. Extraction of slice parts from volumic file extents

The parallel Visible Human slice server application was synthesized using the CAP Computer Aided Parallelization tool [Gennart98] and the library of reusable parallel file system components [Messerli97] which enable building pipelined-parallel applications combining high-performance computing and I/O intensive operations. The parallel server architecture has been interfaced to the Microsoft IIS Web server using the ISAPI protocol. A Java 1.1 applet runs on Web clients and enables users to specify slice position and orientation and generate image slice extraction requests. Replies of the Web server are compressed using the JPEG standard and send back to the Web clients for display. The Web interface is operational at *http://visiblehuman.epfl.ch*.

The parallel slice server application consists of a server interface residing on the Web server PC and of server processes running on the server's parallel PC's. The server interface interprets the slice location and orientation parameters defined by the user and determines the image extents which need to be accessed. It sends to the concerned servers (servers whose disks contain the required extents) the extent reading and image slice part extraction requests. These servers execute the requests and transfer the resulting slice parts to the server interface, which asssembles them into the final displayable image slice (Figure 3).
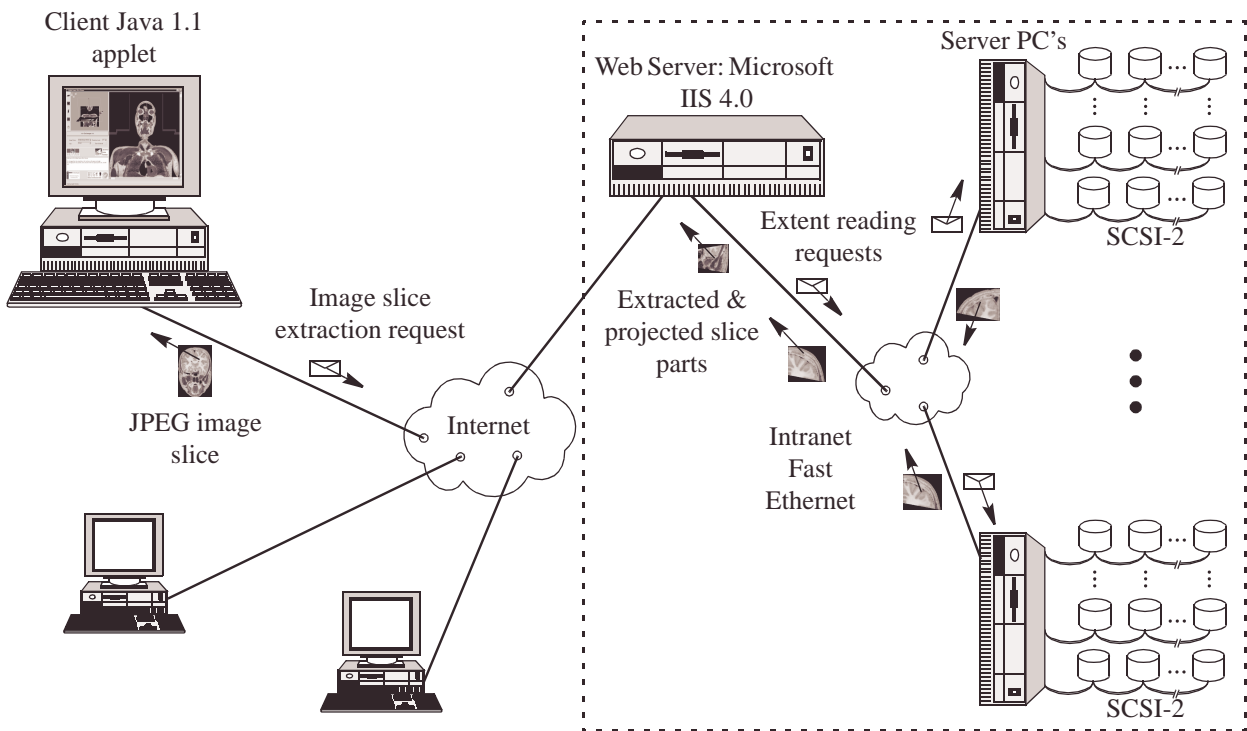

Figure 3. The Visible Human slice Web server configuration

The parallel slice server application is described by the diagram of Figure 4 and by the program of Figure 5. The program of Figure 5 written in the CAP formalism is precompiled into C++ code which is then compiled to run on the server's PC's.
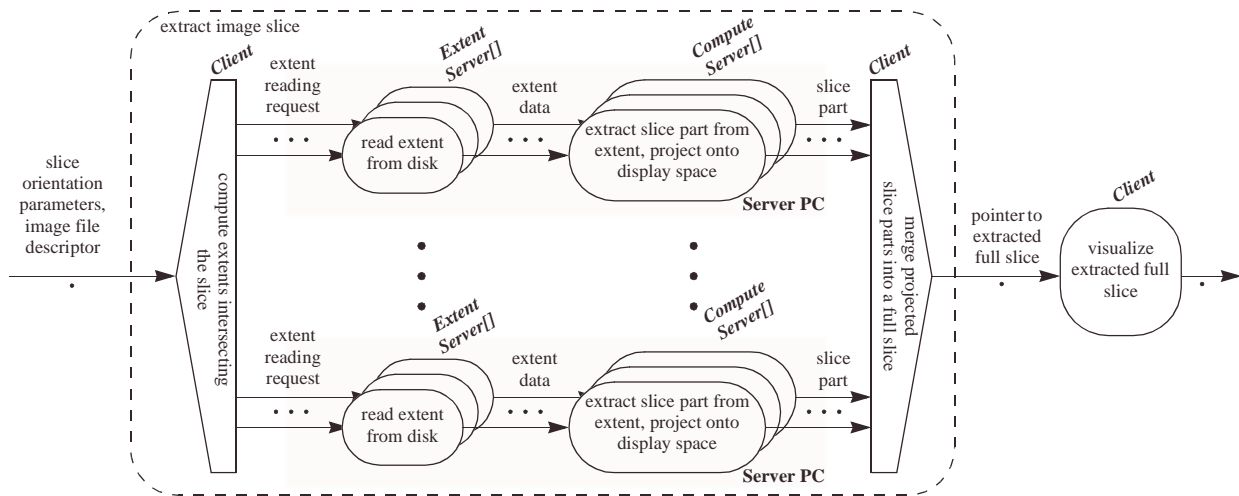

Figure 4. Graphical representation of the pipelined-parallel plane extraction and visualization application

```
1  external leaf operation Ps2ExtentServerT::ReadExtent
2    in ExtentReadingRequestT* InputP
3    out ExtentT* OutputP;
4
5  leaf operation Ps2ComputeServerT::ExtractAndProjectSlicePart
6    in ExtentT* InputP
7    out SlicePartT* OutputP
8  {
9    OutputP = new SlicePartT;
10   ...C++ sequential code
11 }
12
13 bool SplitSliceRequest(SliceExtractionRequestT* FromP,
14                   ExtentReadingRequestT* PreviousP, ExtentReadingRequestT* &ThisP) {
15   ThisP = new ExtentReadingRequestT;
16   ...C++ sequential code
17   return (IsNotLastExtentReadingRequest);
18 }
19
20 void MergeSlicePart(SliceT* IntoP, SlicePartT* ThisP) {
21   ...C++ sequential code
22 }
23
24 operation Ps2ServerT::ExtractSlice
25   in SliceExtractionRequestT* InputP
26   out SliceT* OutputP
27 {
28   parallel while (SplitSliceRequest, MergeSlicePart, Client, SliceT Output)
29   (
30     ExtentServer[thisTokenP->ExtentServerIndex].ReadExtent
31     >->
32     ComputeServer[thisTokenP->ComputeServerIndex].ExtractAndProjectSlicePart
33   };
34 }
35
36 leaf operation ClientT::VisualizeSlice
37   in SliceT* InputP
38   out capTokenT* OutputP
39 {
40   OutputP = new capTokenT;
41   ...C++ sequential code
42 }
43
44 operation Ps2ServerT::ExtractAndVisualizeSlice
45   in SliceExtractionRequestT* InputP
46   out capTokenT* OutputP
47 {
48   Ps2Server.ExtractSlice
49   >->
50   Client.VisualizeSlice;
51 }
```

thread where MergeSlicePart is executed

output token of the parallel while construct

split function *merge function*

Figure 5. CAP specification of the pipelined-parallel plane extraction and visualization application

The server's performance has been tested when striping the Visible Human (male dataset) onto 1 to 5 Bi-Pentium Pro server PC's and onto 1 to 12 disks per server PC (max. 60 disks). Figure 6 shows the number of extracted 512x512 colour slices per second for various configurations.
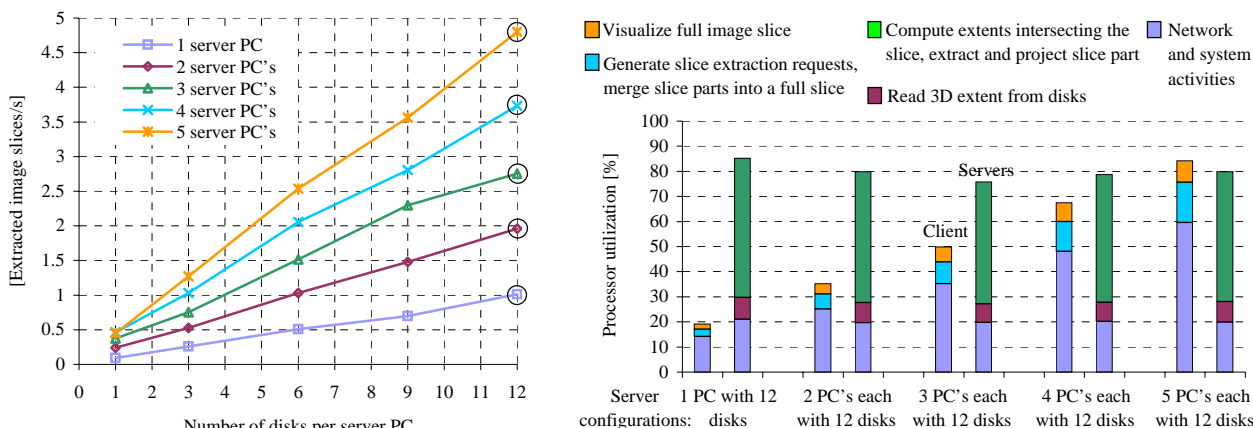

Figure 6. Slice extraction performance under various configurations, without disk caching

Each slice access request is decomposed into 437 volumic extent access requests (22 MBytes). For all the server configurations, disk I/O bandwidth is always the bottleneck (effective single disk throughput for 51KB blocks: 1.88 MBytes/s). With 4.8 image slices/s, the Web server is able to receive from the Fast Ethernet 7.8 MBytes/s of slice parts. These performances are close to the performances offered by the underlying hardware, operating system (Windows NT) and network protocols (TCP/IP).

The CAP computer-aided parallelization tool greatly simplifies the creation of parallel distributed memory server applications. Application programmers create separately the serial program parts and express the parallel behavior of the program with CAP constructs. The parallel part of the program can be easily modified by changing the sequence of operations or by building hierarchical CAP constructs.

References

[Ackerman95] Michael J. Ackerman, "Accessing the Visible Human Project", D-Lib Magazine: The Magazine of the Digital Library Forum, October 1995, http://www.dlib.org/dlib/october95/10ackerman.html

[Gennart98] B. Gennart, M. Mazzariol, V. Messerli, R.D. Hersch, "Synthesizing Parallel Imaging Applications using CAP Computer-Aided Parallelization Tool", IS&T/SPIE's 10th Annual Symposium, Electronic Imaging'98, Storage & Retrieval for Image and Video Database VI, San Jose, California, USA, January 1998, 446-458

[Messerli97] Vincent Messerli, Benoit A. Gennart, Roger D. Hersch, "Performances of the PS2 Parallel Storage and Processing System for Tomographic Image Visualization," *Proceedings of the 1997 International Conference on Parallel and Distributed Systems*, IEEE Computer Society Press, Seoul, Korea, December 1997, 514-522

[North96] C. North, B. Shneiderman, C. Plaisant, "User Controlled Overviews of an Image Library: The Visible Human Explorer," *The Visible Human Project Conference Proceedings*, October 1996, http://www.nlm.nih.gov/research/visible/vhp_conf/vhpconf.htm