

Título del trabajo: Visualización Distribuida Usando Sistemas Operativos Convencionales

Autores: Osvaldo Clúa oclua@acm.org y María Feldgen mfeldgen@ieee.org

Institución: Laboratorio de Sistemas Distribuidos Heterogéneos Facultad de Ingeniería UBA

Resumen: Se desarrolló un ambiente de Visualización distribuida, agregando un algoritmo de control de admisión a un Sistema Operativo de propósito general. Este trabajo describe el escenario de visualización y algunos esquemas de control de admisión con poco *overhead*. Se presenta el algoritmo de predicción de uso de los recursos de red propuesto basado en la envolvente del tráfico y en la carga del sistema. Se discuten los detalles de implantación y los tópicos de visualización asociados. Se presentan algunos resultados experimentales y se analizan conclusiones de alcance general. El algoritmo tuvo un buen desempeño convirtiendo un sistema no usable en uno controlado. Aplicaciones similares pueden aumentar la satisfacción del usuario en varios escenarios multimediales.

Keywords: Redes de Computadoras, Sistema Distribuidos.

VISUALIZACIÓN DISTRIBUIDA USANDO SISTEMAS OPERATIVOS CONVENCIONALES

Oswaldo Clúa
oclua@acm.org

María Feldgen
mfeldgen@ieee.org

Laboratorio de Sistemas Distribuidos Heterogéneos
Facultad de Ingeniería
UBA

Resumen: Se desarrolló un ambiente de Visualización distribuida, agregando un algoritmo de control de admisión a un Sistema Operativo de propósito general. Este trabajo describe el escenario de visualización y algunos esquemas de control de admisión con poco *overhead*. Se presenta el algoritmo de predicción de uso de los recursos de red propuesto basado en la envolvente del tráfico y en la carga del sistema. Se discuten los detalles de implantación y los tópicos de visualización asociados. Se presentan algunos resultados experimentales y se analizan conclusiones de alcance general. El algoritmo tuvo un buen desempeño convirtiendo un sistema no usable en uno controlado. Aplicaciones similares pueden aumentar la satisfacción del usuario en varios escenarios multimediales.

1. Introducción.

La creciente habilidad para modelar y medir una amplia variedad de fenómenos nos ha inundado de datos. Sin embargo, generar y adquirir datos no es un fin en si mismo, sino una forma de obtener conocimiento. A pesar del rol relevante que juega el análisis de datos, en muchas circunstancias solo logramos entender el fenómeno bajo estudio gracias a la habilidad del sistema visual humano de reconocer e interpretar patrones complejos [Foster, 1999]. Esta habilidad se convierte en un coadyuvante de vital importancia para detectar significados y anomalías en conjuntos de datos científicos. Para permitir ver objetos gráficos derivados de los datos recogidos o simulados la visualización es una herramienta importante dando lugar a lo que los filósofos llaman percepción significativa [Dretsake, 1995].

En el proceso de visualización se encuentran cada vez mas a menudo fuentes de datos, usuarios y dispositivos para visualización y análisis distribuidos geográficamente. Estos escenarios de visualización a distancia introducen nuevos problemas. Los sistemas de computación actuales permiten la exploración interactiva de conjunto de datos grandes por medio de dispositivos tales como displays de alta resolución o de inmersión [Leigh,1999], y la aplicación de algoritmos tales como extracción y rastreo de características (*features*) [Globus, 1995]. Como el ataque algorítmico del análisis de datos demanda cada vez mayores recursos, se hace cada vez mas difícil compartir los ambientes de captura (o generación simulada) de datos y de visualización en una sola plataforma o en la misma ubicación física. Es por eso que la distribución de roles entre sistemas de computación heterogeneos comienza a tener lugar en el campo de la visualización científica.

A pesar de que las aplicaciones interconectadas permiten disponer de recursos especialmente adaptados y distribuidos geográficamente para la tarea de visualización, el ambiente de computación distribuida resultante abre una serie de problemas de análisis de desempeño y de optimización. Los ambientes de ejecución de las aplicaciones distribuidas se hacen cada vez menos determinísticos ya que los sistemas corren simultáneamente varias tareas, haciendo variar de una ejecución a otra el ancho de banda y la latencia de las redes. Es importante poder identificar los cuellos de botella del desempeño para mejorar el comportamiento del sistema como un todo.

Si se deben compartir los sistemas de computación y la redes de comunicaciones con otras aplicaciones, se debe prestar especial atención al ancho de banda de la red y a la posibilidad del servidor de mantener una corriente continua de datos a visualizar (*data streaming*). En sistemas de tiempo real, la técnica usada por el servidor para comprometer y mantener una corriente continua de medios continuos tales como video se conoce como control de admisión [Xu, 1993]. En este trabajo nos concentraremos en un

sistema de control de admisión para un servidor de visualización lo suficientemente simple como para implementarse en un Sistema Operativo de uso corriente tal como UNIX.

2. El escenario de Visualización Remota.

En nuestro proyecto de visualización, detallado al final del trabajo, elegimos como software de visualización a Vis5D. Está disponible en <ftp://www.ssec.wisc.edu/pub/vis5d-5.2>. Vis5D fue desarrollado por la University of Wisconsin-Madison Space Science and Engineering Center (SSEC) Visualization project por Bill Hibbard, Johan Kellum y Brian Pau. En <http://www.ssec.wisc.edu/billh/vis.html> están los detalles sobre su uso. Hace uso del sistema X Window como display gráfico y puede aceptar *streams* de datos para representar desde un *pipe* o un *network stream* (puede ser usado además como *helper* de algún *browser*) Su sistema de seguimiento de características y sus capacidades de animación (en realidad FVis5D en <http://www.caip.rutgers.edu/~xswang/vis5d>) pueden usarse para crear un *stream* MPEG [le Gall, 1991].

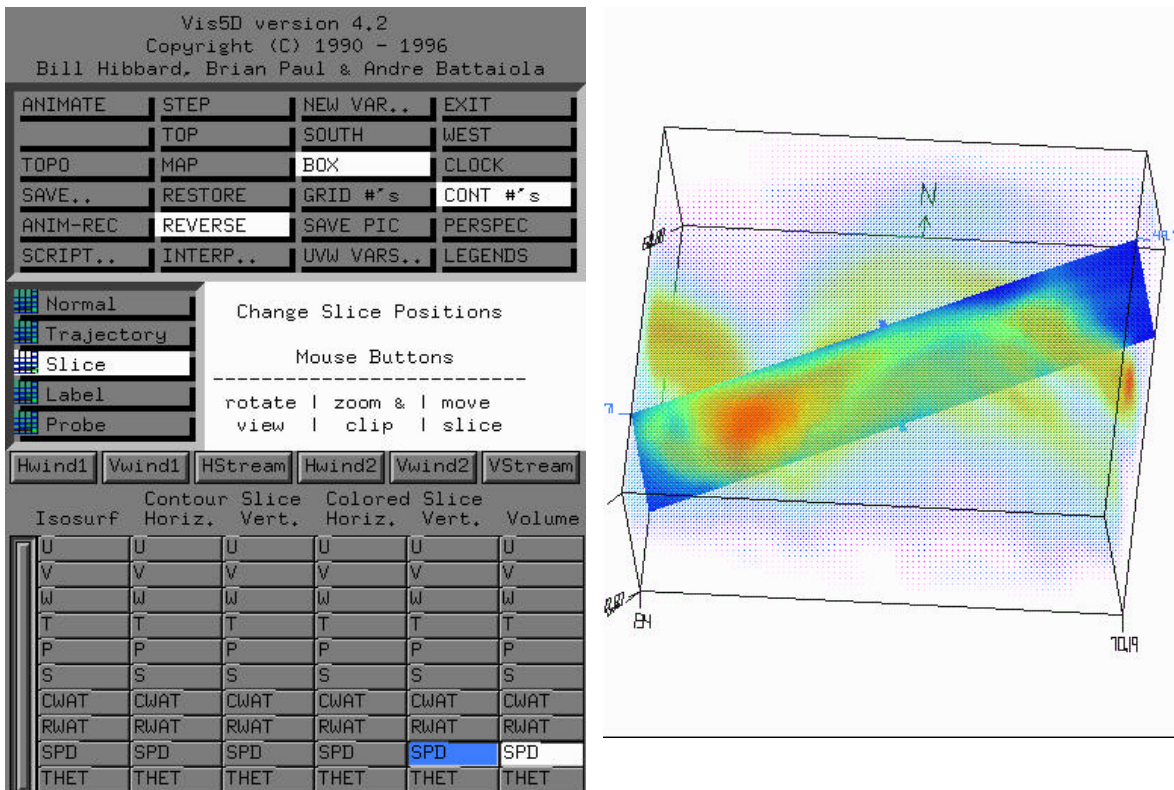


Figura 1. Controles y ventana de visualización de Vis5D

Los componentes críticos en el diseño de los servicios de *streaming* son los servers, que almacenan y recuperan o generan los datos y el subsistema de red que entrega sincrónicamente la información a los clientes. El caso general de *streaming* en una red con demoras distribuidas según una distribución gaussiana se analizó en [Clúa 1998a]. Pueden encontrarse tópicos relacionados con almacenamiento y recuperación en [Clúa, 1998b] y en sus referencias.

Identificamos tres formas distintas de *streaming* en nuestro escenario:

- Un *stream* MPEG off line generado y previamente almacenado.
- El envío de los datos a visualizar por un *pipe*.
- Un stream gráfico generado por el Sistema X Window.

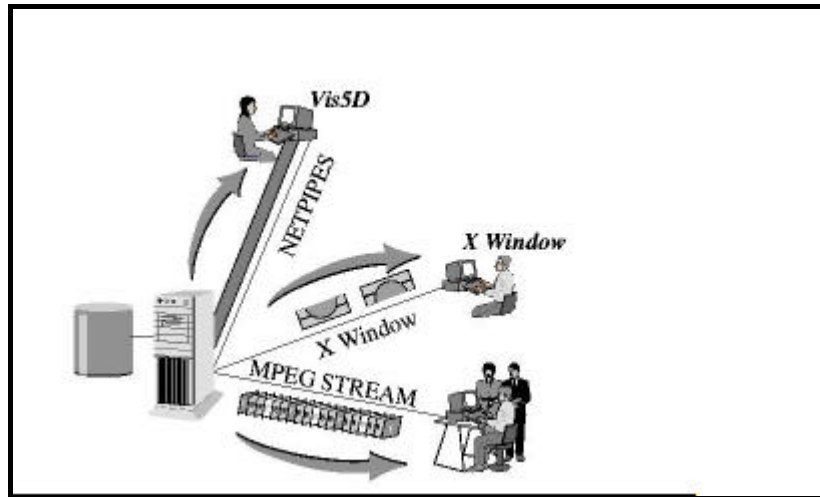


Figura 2 Escenarios de *streaming*

Cada uno de estos escenarios se refieren a una tarea de visualización diferente, y los tres se usaron en nuestro proyecto. Por ello analizamos soluciones generales al problema de control de admisión factibles para estas tres instancias. Para no restringir nuestro trabajo, supusimos que se le puede exigir al servidor varias tareas distintas de *streaming* simultáneamente y para varios clientes.

3. Características de los Servidores de Medios Continuos.

Si bien no se dispone de las facilidades de un sistema de tiempo real, un análisis de las técnicas más sencillas de control de admisión usadas en servidores de tiempo real nos permiten describir y fundamentar el método adoptado.

La reproducción de un *stream* continuo es una secuencia de tareas periódicas con vencimientos. Las tareas corresponden a la recuperación de los datos y los vencimientos a los tiempos planificados de reproducción. A pesar que es posible que un sistema multimedial pueda recuperar en forma continua las unidades de datos *just in time* para ser reproducido, en la práctica las tareas de recuperación se distribuyen en forma de ráfagas. Los bloques de datos deben almacenarse en *buffers* cuando la recuperación adelanta a la reproducción y el server debe proveer a los buffers con datos suficientes como para asegurar que la reproducción no los agote.

Para cumplir con requerimientos múltiples, el procesamiento se hace en rondas. Esto tiene la ventaja de mantener los algoritmos sincronizados con las técnicas de exploración del disco [Anderson, 1992]. Durante una ronda, la cantidad de datos recuperados para un *stream* es al menos igual a la cantidad de datos consumidos por la reproducción de ese *stream*. Esto significa que, ronda a ronda, la producción nunca deja atrás al consumo y que nunca hay un decrecimiento neto de la cantidad de datos en el buffer. Los algoritmos con esta propiedad se conocen como *buffer-conserving algorithms* [Gemmel, 1994].

Una vez conocidas las restricciones de performance de tiempo real, un servidor multimedia debe emplear control de admisión para determinar si puede dar servicio a un nuevo *stream* sin afectar las tareas en curso. Por tratarse de medios continuos, pueden tolerarse algunas pérdidas de vencimientos. Al hacer *streaming* de datos para su visualización, la situación es más comprometida que en el caso de multimedia, ya que hay que prever que el software de visualización requiere una cantidad mínima de datos para producir su salida.

Un servidor puede brindar tres tipos generales de calidad de servicio [Vin, 1993] al decidir si admite o no un *stream* adicional:

- *Determinístico*. se garantiza cumplir con todos los vencimientos, en general se consideran los escenarios más desfavorables para el algoritmo de admisión.
- *Estadístico*. Se garantiza una cierta probabilidad de cumplir con los vencimientos
- *Background*. No hay garantía para los vencimientos. El servidor planifica estos accesos cuando hay tiempo libre dejados por los servicios de las categorías anteriores.

Para implementar un servicio determinístico, los recursos se reservan según la técnica del peor caso: antes de admitir otro *stream* y alargar la duración de la ronda, el server debe asegurar el *buffering* necesario para impedir el agotamiento de alguna reproducción.

El servicio estadístico se parece al determinístico, pero los cálculos del alargamiento de la ronda se basan en valores estadísticos.

4. Algoritmos de Control de Admisión Basados en Ranuras de Tiempo (*slices*).

Con el objeto de reducir la complejidad inherente a los algoritmos de admisión estadísticos, [Xiaoye, 1999] propone dos técnicas basadas en ranuras de tiempo o *slices*. La estimación del Ancho de Banda requerido real de un *stream* $BW(t)$ puede hacerse de varias formas. El control de admisión determinístico se obtiene reservando el máximo Ancho de Banda (BW_{max}) del *stream*. Por supuesto, esto requiere conocimiento previo de los requerimientos de Ancho de Banda de la secuencia a transmitir; lo que se logra únicamente en el caso de una secuencia MPEG ya almacenada. Este valor servirá como punto de comparación.

Si el tiempo de reproducción total de una secuencia es T , un esquema de *slicing* divide $[0, T]$ en varios intervalos ts . El máximo valor de ts es T y el mínimo es una *unidad de tiempo* tu que puede corresponder a una ronda del server o ser menor que ella. En el caso determinístico, el BW_{max} para un ts , obtenido al dividir T en $n*ts$ intervalos genéricos i es

$$BW_{max}(ts) = \max\{BW(i)\}; (i=0, 1, \dots, n; n=T/ts)$$

Ecuación 1.

El server acepta el pedido si se puede satisfacer el ancho de banda total, sumando el BW_{max} para todos los *streams*. Esta decisión no requiere mas que un preproceso que no impone mucho consumo de recursos y almacenar los valores para un ts dado.

Otra forma de realizar el cálculo es usar solo el BW_{max} de la secuencia remanente aún no transmitida, o Máximo Futuro determinado como

$$BW_{max}(ts) = \max\{BW(i)\}; (i=i, i+1, \dots, n; n=T/ts)$$

Ecuación 2.

Esta forma también requiere de un preproceso y, al llegar un nuevo pedido, se usa el máximo futuro de todas las secuencias en transmisión para determinar si puede o no aceptarse. Los detalles de implementación se desarrollan en el trabajo de [Xiaoye, 1999] ya mencionado.

5. Algoritmo de Admisión por Promedios Ponderados propuesto.

En uno de nuestros escenarios, basados en sistemas operativos sin facilidades de tiempo real, se solicita el *streaming* de una secuencia MPEG previamente almacenada. En los otros, se transmiten los datos a visualizar o un *stream* del sistema X Window. En el caso de enviar los datos, la magnitud del Ancho de Banda requerido dependerá de la fuente de datos y en el caso del sistema X Window se implementa el protocolo descrito en el RFC 1013 o bien algún derivado de **dxpc** de Zachary Vonler (ver *man dxpc (1)*). En ambos casos no hay acceso previo a la secuencia forzando el uso de un algoritmo estadístico predictivo.

En el caso de MPEG, las restricciones están impuestas por el medio continuo. El usuario espera ver una animación donde el tiempo *corre* a velocidad constante y podemos asumir que es el mas exigente de los tres casos. Como la secuencia MPEG está almacenada con anticipación, se la preprocesa para adjuntarle un registro con el valor del Máximo Futuro para cada ranura de tiempo ts . Si se cambia este valor, se debe reprocesar la secuencia, pero la operación no requiere de mucho tiempo ni añade mayor exigencia de almacenamiento.

En los escenarios restantes, el usuario manipula la interface para obtener la representación de la visualización con los controles ilustrados en la Figura 1. En un caso la fuente debe enviar los datos y el otro se reduce a una operación de interface remota. Como el usuario tiene un cierto control sobre los requerimientos, no es tan importante tener una respuesta continua. Si parece necesario brindar claves visuales de la actividad de comunicación y hay muchas de estas aplicaciones disponibles para X Window tales como el *KDE Desktop Environment* o el popular **nettop**. El software de visualización Vis5D presenta además información sobre los datos leídos como realimentación de la operación de entrada al pipe, que se

despliegan en la interface del usuario. Si bien estos controles nos permiten ser mas permisivos con el control de admisión, la carga del server debe limitarse en forma tal de mantener al sistema en condiciones de uso.

Para evaluar la carga cuando llega un nuevo pedido, proponemos un peso ponderado de las rondas anteriores en una mezcla como la usada en el cálculo del *round trip time* de TCP/IP en el RFC 793 [Postel, 1981]

$$BW_{max}(t+1) = \alpha * BW_{max}(t) + (1-\alpha) * BW_{maxEst}(t+1)$$

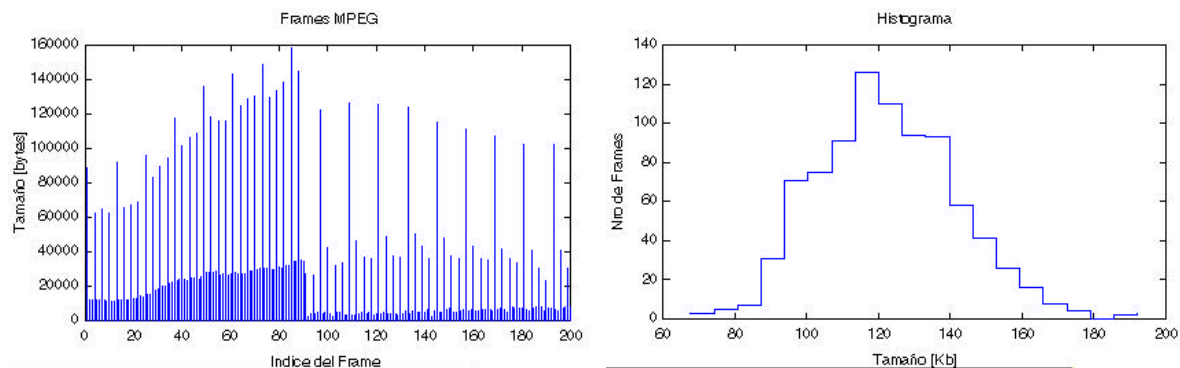
Ecuación 3

El término $BW_{maxEst}(t+1)$ es el estimado que se corrige usando α para controlar el peso de la historia en la predicción. Los valores de BW se obtienen del máximo futuro de las secuencias de MPEG en reproducción y del sistema de registro de la red para el resto de las comunicaciones.

La administración de la carga de proceso del *server* se desarrolla en la próxima sección, ya que es altamente dependiente del sistema.

6. Resultados Experimentales.

Se usó un sistema no totalmente dedicado con una red TCP/IP y varios sistemas operativos heterogéneos como Linux y OS/2 en máquinas con otras aplicaciones activas. No se usaron las facilidades de tiempo real de ninguno de los dos Sistemas Operativos por no tratarse de máquinas dedicadas. Además se usó la versión *standard* XFREE 86 3.3.6 del sistema X Window. Usamos tres secuencias MPEG pregrabadas obtenidas al visualizar diferentes aspectos de nuestro proyecto. Uno de los histogramas de los tamaños típicos de los frames "I" se muestra en la Figura 3, al lado del detalle de la secuencia MPEG durante un cambio de escena.



**Figura 3 Izquierda: Perfil de Frames de una secuencia MPEG (dos escenas)
Derecha: Histograma de los tamaños de los Frames I**

Preprocesamos las secuencias y almacenamos los valores de los máximos futuros BW_{max} para una ranura de tiempo que corresponde a un Grupo de Cuadros MPEG (*Group of Pictures, GOP*). Es el valor de detalle que parece mas natural para una ranura de tiempo. La distribución de los BW_{max} futuros se ve en la figura 4 comparada con los requerimientos exactos y el máximo absoluto que corresponde al caso de control de admisión determinístico.

Se impuso una carga máxima del 50% al server, permitiendo una ligera sobrecarga. Esto se debe a no disponer de máquinas dedicadas y a utilizar Sistemas Operativos Convencionales. Un cálculo conservador del factor de cola para una carga del 50% indica que el sistema estaría utilizado en ocasiones el doble (100%) [Dowdy,1993]. De haber usado las facilidades de Tiempo Real, se podría haber implementado un algoritmo tal como EDF [Buttazzo, 1997] o algún otro para obtener un punto de trabajo promedio mas alto. Además se impuso un máximo al Ancho de Banda destinado a las aplicaciones de visualización.

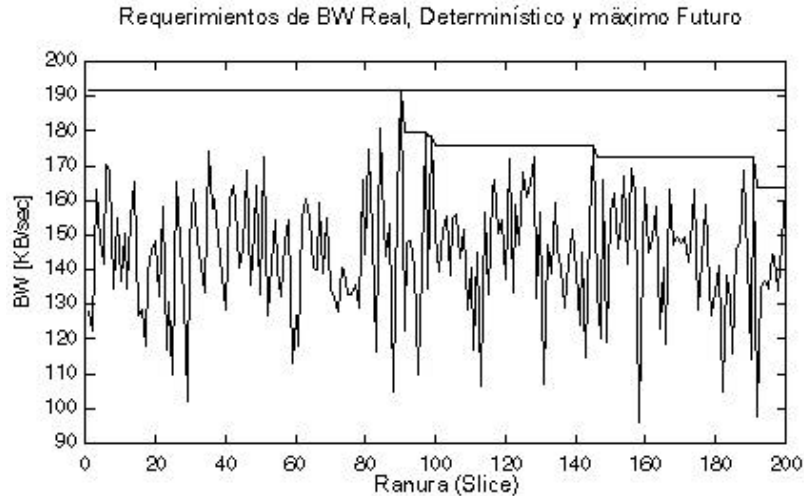


Figura 4: BW de la secuencia MPEG “ranurada”, Demanda Total Máxima, Máximo BW Futuro y Demanda Real.

En el caso de un pedido remoto como “cat IN035 | rsh lechuza.fi.uba.ar Vis5D” el perfil de transferencia indica una gran actividad al principio (la grilla, topografía, controles y fondo de visualización) seguida por la actividad correspondiente a los propios datos a visualizar, cuya velocidad queda limitada por la fuente de datos. Usamos un *shell script* para encapsular el pedido y hacer allí el control de admisión. En los pedidos de un *display* remoto usando el sistema *X Window* encontramos un patrón de transferencia similar, con una gran demanda de Ancho de Banda al principio, cuando los Widgets y los datos del fondo y de la ventana de control se transmiten y luego el uso de Ancho de Banda sigue la operación de la visualización. En ambos casos, el requerimiento de Ancho de Banda resulta despreciable en comparación con el *streaming* MPEG.

Cuando llega un pedido, el control de admisión se evalúa de la siguiente manera:

- Si es un pedido interactivo, se lo acepta si la carga del sistema es menor que el límite impuesto. (El valor del 50% permitió correr aplicaciones de oficina en conjunto, facilitando la redacción de trabajos relacionados).
- Si es un pedido de *streaming* se evalúa el Ancho de Banda Futuro para todo el tráfico MPEG ($BW_{maxEst}(t+1)$), se calcula el promedio suavizado según la ecuación 3 y se le suma el Ancho de Banda Futuro del *stream* requerido. Este valor se compara con el Ancho de Banda de la red (o la porción destinada a estas aplicaciones). Si es menor se acepta el pedido, sino se lo rechaza.

El uso de la red y la carga del sistema se toman de los monitores del sistema. No tuvimos por objetivo obtener valores precisos de estas estimaciones, sino proveer de un cierto control de admisión a un sistema que no proporcionaba ninguno. Los valores no son importantes siempre y cuando se los usen coherentemente.

Los resultados demuestran que el sistema es usable y que puede ajustarse variando los máximos. No intentamos hacerlo en forma controlada o dinámica, nos limitamos a encontrar un punto de trabajo por encima del cual el sistema dejaba de ser usable. Como las evaluaciones de carga y red son altamente dependientes de nuestro sistema (y de las otras aplicaciones que se usan simultáneamente), las cifras usadas tienen sentido solamente en ese contexto. Incluso, usando diferentes máquinas y reservado distintos anchos de banda para la misma mezcla de tareas, estas eran aceptadas o rechazadas al representar cargas distintas, como era de esperarse.

7. Conclusiones.

El control de admisión puede agregarse a los sistemas operativos de propósito general y el esfuerzo rinde buenos frutos. Las estrategias propuestas son lo suficientemente simples y de baja complejidad como para intentarlo siempre que los Sistemas operativos brinden las facilidades de base necesarias como es el caso de UNIX y OS/2. En aplicaciones de visualización distribuida este control puede resultar en la diferencia entre un sistema usable y uno que no lo es. Al admitir la coexistencia con aplicaciones generales

las tareas de captura de pantalla, reportes y análisis de resultados se agilitan notablemente. Si un sistema de visualización se debe incluir en un lazo de control abierto con intervención humana, puede refinarse más estudiando los resultados de las operaciones interactivas y previendo alternativas tales como sacrificar la resolución de una aplicación en favor de un nuevo requerimiento. Debe prestarse especial atención al uso de textura y color para representar valores, ya que el Sistema Visual Humano los procesa en forma muy pobre como para confiar en ellos [Clúa 2000]. Es preferible usar herramientas de presentación focalizada y de medición remota.

Este proyecto es parte de un proyecto trienal codificado como IN035 "Problemas Inversos en la Conducción del Calor, Desarrollo de Algoritmos y Aplicaciones Tecnológicas" de la Secretaría de Ciencia y Técnica de la Universidad de Buenos Aires.

References.

- [Anderson, 1992] **D. Anderson, Y. Osawa, and R. Govindan,** "A File System for Continuous Media," *ACM Trans. Computer Systems*, Vol. 10, No. 4, Nov. 1992, pp. 311-337.
- [Buttazzo, 1997] **G. Buttazzo:** *Hard Real Time Computing Systems: Predictable Scheduling Algorithms and Applications*, Kluwer, 1997.
- [Clúa 200] **O. Clúa; M. Feldgen:** "textura y color en representación focalizada", accepted for publication in *ICIEY2K*, Facultad de Ingeniería, UBA, Buenos Aires, 2000. (CD-ROM Proceedings)
- [Clúa, 1998a] **O. Clúa; M. Feldgen:** "Compensación Adaptiva de Buffers en Multimedia Distribuída", *Anales de REDES98, IX CLAIO, 27JAIIO*, Facultad de Ingeniería, UBA, Buenos Aires, 1998, pp 9 - 16.
- [Clúa, 1998b] **O. Clúa; M. Feldgen:** "Requerimientos para un Sistema de Archivos de Medios Continuos", *Anales de ICIE98*, Facultad de Ingeniería, UBA, Buenos Aires, 1998, pp185 - 188.
- [Dretsake, 1995] **F. Dretsake:** "Meaningful perception", in *An Invitation to Cognitive Science: Visual Cognition* MIT Press, 1995, pp 331-352.
- [Dowdy,1993] **L. Dowdy, C. Lowery:** *Performance Supplement to Operating Systems*, Prentice Hall, 1993, pp53 y ss.
- [Foster 1999] **I. Foster; J. Insley; G. Laszavsky; C. Kesselman; M. Thiebaux,:** "Distance Visualization" in *IEEE Computer*, Dec. 1999, p36-43.
- [Gemmell, 1994] **D.J. Gemmell, and J. Han,** "Multimedia Network File Servers: Multichannel Delay Sensitive Data Retrieval," *Multimedia Systems*, Vol. 1, No. 6, Apr. 1994, pp.240-252.
- [Globus, 1995] **A. Globus; S. Uselton:** "Evaluation of Visualization Software" Report NAS-95-005 Computer Sciences Corporation at NASA Aames Research Center.
- [Kleinrock, 1976] **L. Kleinrock:** *Queueing Systems, Vol II: Computer Applications*, Wiley, 1976.
- [le Gall, 1991] **D. le Gall.:** "MPEG: A video compression standard for multimedia applications", *Communications of the ACM*, Vol 34, 4, Abril 1991, pp 46-59.
- [Leigh, 1999] **J. Leigh:** "Visualization in Teleimmersive Environments" in *IEEE Computer*, Dec. 1999, pp. 66-73.
- [Postel, 1981] **J. Postel:** *RFC 793, Transmission Control Protocol*, 1981
- [Vin, 1993] **H.M. Vin, and P. Venkat Rangan,** "Designing a Multi-User HDTV Storage Server," *IEEE J. Selected Areas in Comm.*, Vol. 11, No. 1, Jan. 1993,pp 153-164.
- [Xiaoye, 1999] **Jiang Xiaoye, P. Mohapatra:** "Efficient admission control algorithms for multimedia servers", in *ACM Multimedia Systems* Vol 7 Nr 4, pp 295-304, 1999.
- [Xu, 1993] **J. Xu.; D. L. Parnas:** "On satisfying constraints in Real Time Systems" *IEEE Transactions on Software Engineering* 19(1): January 1993, pp. 70-84.