

# Visualización del Balance de Carga en un Sistema Distribuido<sup>1</sup>

III Workshop de Computación Gráfica, Imágenes y Visualización (WCGIV)

Martín Larrea<sup>1</sup>, Sergio Martig<sup>1</sup>, Silvia Castro<sup>1</sup>, Javier Echaiz<sup>2</sup>

<sup>1</sup>VyGLab, Laboratorio de Visualización y Computación Gráfica

<sup>2</sup>LISiDi, Laboratorio de Investigación en Sistemas Distribuidos

Dpto. de Ciencias e Ingeniería de la Computación

Universidad Nacional del Sur

Avda. Alem 1253

8000 – Bahía Blanca, Argentina

{mll, srm, smc, je}@cs.uns.edu.ar

## Resumen

Balancear la carga de un sistema distribuido involucra equilibrar la distribución de tareas en el mismo con el objetivo de lograr un aumento de rendimiento. La principal herramienta utilizada para esto es la migración de procesos o de datos; ésta debe hacerse rápidamente para no interferir con el aumento de rendimiento y procesamiento total buscados. Para esto resulta de gran importancia contar con una herramienta que permita visualizar el estado de carga de un sistema distribuido. Desde el campo de la Visualización de información y el estudio de la Interacción Hombre-Computadora se pueden obtener herramientas y técnicas que provean una visualización clara y eficiente de estos datos teniendo en cuenta las capacidades del hombre. El objetivo de este trabajo es combinar el trabajo realizado desde la Visualización de Información y la Interacción Hombre-Máquina con la visualización del balance de carga en un sistema distribuido para obtener una herramienta que permita visualizar determinados parámetros de un sistema distribuido; así a simple vista se podrá tener una visión del comportamiento del estado de carga del sistema, tanto a nivel de la aplicación distribuida como de cada uno de sus componentes.

**Palabras Clave:** Visualización de Información - Balance de Carga – Sistemas Distribuidos

## 1. Introducción

Un algoritmo de balance de carga trata de equilibrar la carga total de un sistema transfiriendo en forma transparente la carga de trabajo de nodos con cargas pesadas a nodos más descargados, aumentando el paralelismo de ejecución lo que lleva al lógico aumento del rendimiento y del procesamiento total. La principal herramienta que se utiliza para este objetivo es la migración de procesos o de datos.

La migración de procesos involucra una serie de toma de decisiones que deben ser llevadas a cabo en forma rápida y ajustada de modo de no interferir con el aumento de rendimiento y de procesamiento total que se busca. Se debe determinar qué tareas deben migrarse, cuándo deben migrarse y dónde deben migrarse, considerando los costos de las distintas alternativas en cuanto a procesamiento y a comunicación.

Estos objetivos derivan en un sinnúmero de problemas abiertos o que tienen soluciones parciales adecuadas a una determinada aplicación. El problema más relevante lo constituye el encontrar una posible métrica que permita tomar las decisiones necesarias para lograr el balance de carga.

Este trabajo surge entonces como una necesidad de contar con una herramienta que permita visualizar el estado de carga de un sistema distribuido. Para alcanzar este objetivo se desarrollaron técnicas que permiten visualizar determinados parámetros de un sistema distribuido, de manera que a simple vista

---

<sup>1</sup> El presente trabajo fue parcialmente financiado por la Universidad Nacional del Sur, Bahía Blanca, Argentina.

sea posible obtener una idea del comportamiento del estado de carga del sistema, tanto a nivel de la aplicación distribuida como de cada uno de sus componentes.

A partir de esta visualización, el usuario experto podrá tomar las acciones que considere adecuadas para sintonizar el sistema o modificar el algoritmo. Si considera conveniente que para estabilizar la carga del sistema, por ejemplo, debe migrar procesos, la visualización lo asistirá en dicha toma de decisión. De este modo, el contar con una herramienta gráfica que permita visualizar el estado del sistema distribuido ayudará a tomar decisiones y permitirá una mejor y más rápida comparación del estado de los distintos nodos involucrados. Se continúa lo desarrollado en [1] presentando una nueva técnica de visualización y se extiende el conjunto de parámetros a visualizar agregando el layout o topología del sistema distribuido. Este trabajo surge en el marco del Dpto. de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca como un trabajo interdisciplinario entre los laboratorios de Investigación en Visualización y Computación Gráfica (VyGLab) y del Laboratorio de Investigación en Sistemas Distribuidos (LISiDi).

En la próxima sección se presenta cuál es el problema de balance de carga que se pretende ayudar a resolver mediante herramientas visuales y se resaltan las restricciones asumidas; luego se describe cada uno de los componentes de la visualización y se detalla cómo se integran para formar una vista del estado de carga del sistema. Finalmente se dan las conclusiones obtenidas y los desarrollos en curso.

## **2. El problema de balance de carga**

El balance de carga constituye una herramienta teórica sumamente adecuada para lograr mayor rendimiento y procesamiento total del sistema, pero en la práctica plantea problemas que no son inmediatamente solubles dadas todas las restricciones que fueron enumeradas.

Los elementos más importantes a considerar cuando se pretende compartir la carga son:

- Determinar hacia dónde enviar una tarea. Las opciones más comunes involucran prueba, multicasting o centralizado. El método de prueba consiste en enviar una solicitud a varias máquinas para determinar si está disponible alguna con baja carga. Los métodos para mantener información centralizada de la carga y multicasting suelen ser costosos y no siempre escalan bien. La decisión en este caso fue la elección de un método de prueba guiado por la información que se tiene sobre el estado actual del sistema.
- Definir qué se entiende por carga alta y qué por carga liviana. Esto está fuertemente ligado con determinar cuáles son los parámetros del sistema que deben monitorearse para poder determinar el nivel de carga. Algunos ejemplos consideran la cantidad de procesos esperando por la CPU, el porcentaje ocioso de la máquina, etc. En este trabajo, las métricas que se consideraron de interés para poder estimar la carga de cada una de las máquinas comprenden, la capacidad de memoria de la máquina y la memoria disponible, el porcentaje de utilización de la CPU, la cantidad de procesos corriendo en cada una de las estaciones y la cantidad y volumen de los mensajes entrantes y salientes.

Una característica que tiene este problema es que no se puede predecir a priori cuándo el sistema se encontrará no balanceado. La visualización estará siempre disponible para que los usuarios tomen la

información que necesiten en cualquier momento; ante una situación de inestabilidad, tampoco será el sistema el responsable de la toma de decisión. La técnica de visualización que se presenta en este trabajo no ofrece características que permitan predecir situaciones que requieran la necesidad de tomar la decisión, ni tomarán decisiones. Este rol será totalmente asumido por los usuarios.

### **3. Marco del problema**

De acuerdo a lo expuesto, el objetivo consiste en visualizar la carga de un sistema distribuido en función de los parámetros establecidos. El sistema está conformado por N nodos conectados entre si mediante una red. Cada nodo está caracterizado por un número que lo identifica, un tamaño de memoria y los porcentajes de CPU ocupada y disponible. En cada nodo se ejecutan hasta M procesos, con  $M > 1$ . La información de cada proceso corresponde a la memoria que usa el proceso, la cantidad de archivos abiertos, la cantidad de mensajes entrantes y salientes que tiene el proceso y el tamaño promedio de los mismos.

Estos son los parámetros que se establecieron para visualizar con el objetivo de poder determinar cuáles son las máquinas con carga alta y cuáles las más ociosas. Se asume que todos estos datos se informan a la visualización.

Se asume que, de acuerdo a lo planteado, el sistema de visualización no tomará decisiones propias, ni propondrá alternativas; serán los usuarios quienes asumirán este rol a partir de la información que surja de la visualización. El sistema se limitará a ofrecer ayuda visual para la toma de decisión. Cuando los usuarios así lo requieran podrán, a partir de la visualización, identificar los problemas y, si lo consideran oportuno, ejecutar las acciones pertinentes.

### **4. Descripción de la visualización**

Los indicadores de carga del sistema distribuido sugeridos para ser visualizados pueden ser divididos en:

1. Parámetros generales del nodo:
  - a. Tamaño de la memoria.
  - b. Velocidad de procesamiento del CPU
  - c. Porcentaje de ocupación de la CPU
  - d. Memoria en uso.
2. Parámetros a nivel de cada uno de los procesos en un nodo
  - a. Tamaño en memoria de cada proceso
  - b. Cantidad de archivos abiertos
  - c. Cantidad de mensajes enviados
  - d. Volumen de información enviada
  - e. Cantidad de mensajes recibidos
  - f. Volumen de la información recibida
3. Topología del sistema distribuido.

A continuación se detallan distintos aspectos presentes en la visualización, analizando los diferentes indicadores y su representación visual. Finalmente, los elementos visuales descritos se integrarán para

formar una vista del estado de carga del sistema que permitirá tener una visión de su comportamiento, tanto a nivel de la aplicación distribuida como de cada uno de sus componentes.

#### 4.1. Parámetros generales del nodo

##### Tamaño de la memoria

Para representar gráficamente el tamaño de la memoria de un nodo se utiliza un prisma cuadrado cuya altura es fija y la longitud del lado de la base representa el tamaño de la memoria. Para poder representar los diferentes tamaños de memoria presentes en el sistema la longitud de la base del prisma se encuentra normalizada con respecto a la máxima cantidad de memoria presente en el cluster. Es decir, un nodo con 256 MB de memoria tendrá un prisma asociado, de lado del doble de longitud con respecto al del prisma asociado a un nodo con 128 MB de memoria.

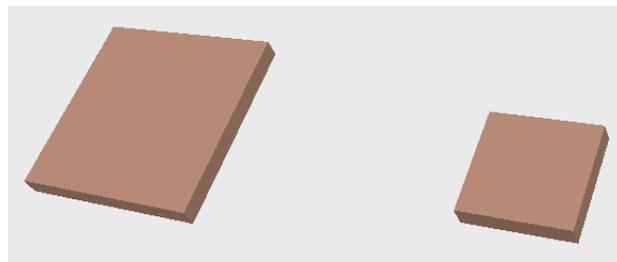


Figura 1. El prisma de la izquierda representa el doble de memoria que el derecho.

##### Velocidad de procesamiento del CPU

La velocidad de procesamiento del CPU presente en el nodo se mapea a la altura de un nuevo prisma cuadrado. Para poder representar las diferentes velocidades de procesamiento presentes en el cluster la altura del prisma se encuentra normalizada con respecto a la máxima velocidad del CPU presente.

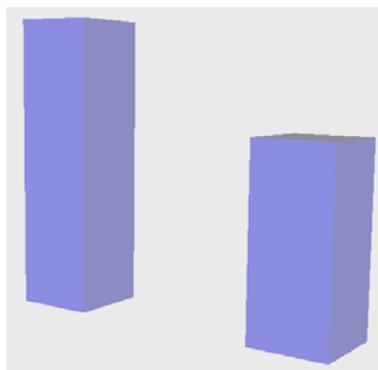


Figura 2. El indicador visual de la izquierda representa una CPU con mayor velocidad de procesamiento que el derecho.

##### Ocupación de la CPU

Como parte de la visualización de este parámetro, debido a que el valor de ocupación de la CPU es un indicador porcentual, su rango de valores se encuentra acotado. El indicador visual de velocidad de procesamiento será representado con un alto nivel de transparencia y dentro del mismo se presenta un segundo prisma cuadrado opaco que indica el valor actual de ocupación de la CPU. Este nuevo prisma posee la misma base que el primero y su altura se encuentra acotada por la altura del indicador de velocidad. Cuando el prisma opaco alcance la altura del prisma transparente se estará representando un nivel de ocupación del 100% y cuando no se visualice el prisma opaco, es decir que tenga altura 0, el nivel de ocupación será del 0%. Cualquier valor de altura intermedio será proporcional al nivel de ocupación de la CPU.

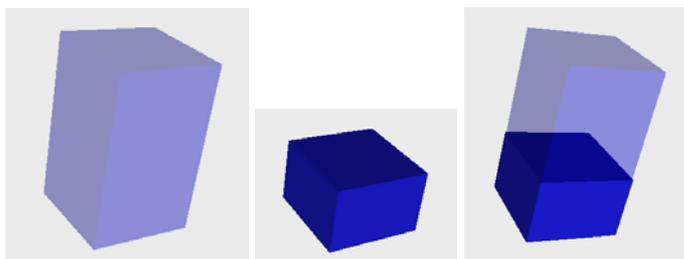


Figura 3. La composición de dos prismas permite establecer el nivel de ocupación de la CPU.

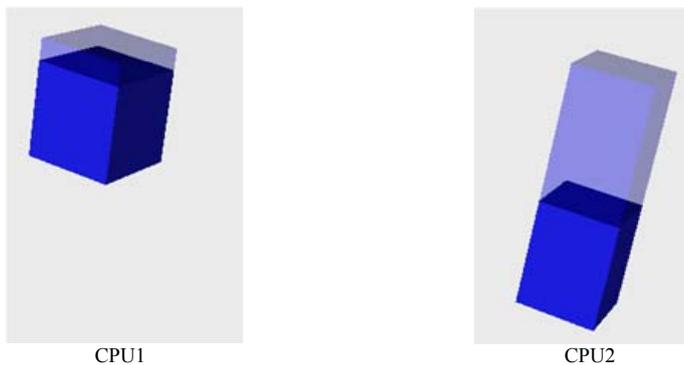


Figura 4. El CPU1 se encuentra casi al 100% de ocupación mientras que el CPU2 esta al 50% de ocupación duplicando la velocidad CPU1.

Los dos indicadores antes mencionados se combinarán para formar una sola figura.

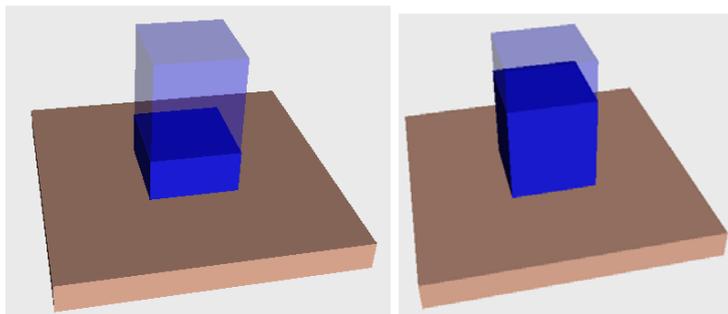


Figura 5. Representación de un nodo.

### Memoria en uso

El área que cubra el indicador de uso de CPU sobre el indicador del tamaño de memoria será el indicador de uso de memoria por parte de la CPU. El área de la base del indicador de uso y velocidad de CPU aumentará o disminuirá en forma proporcional de acuerdo al porcentaje de memoria utilizado por el CPU; el área de la base correspondiente al indicador de tamaño de memoria representa el 100% de la memoria física presente en el nodo.

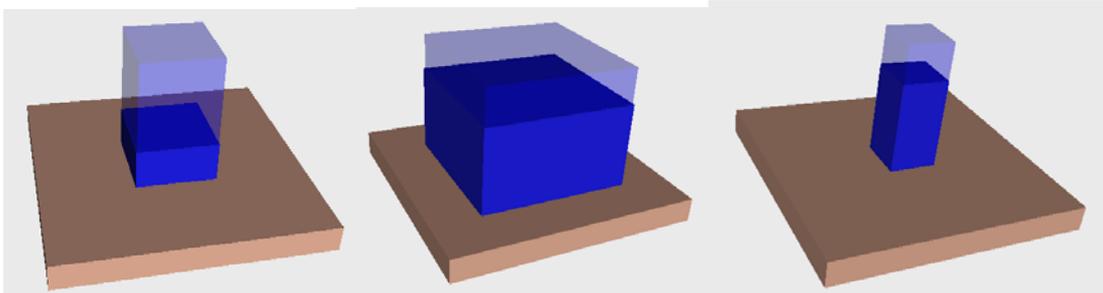


Figura 6. Indicación del uso de memoria por parte del CPU.

En muchos casos el uso de memoria por parte del CPU excede la cantidad de memoria física presente en el nodo y se utiliza entonces memoria virtual. Si se permite aumentar el tamaño del indicador de uso del CPU por sobre el indicador de tamaño de memoria se generaría oclusión como podemos ver en la figura 7.

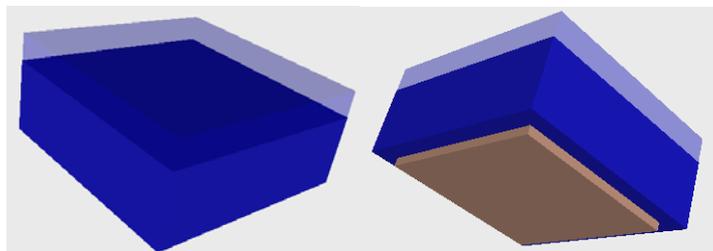


Figura 7. Al aumentar el uso de memoria por parte del CPU se puede generar oclusión sobre el indicador de tamaño de memoria.

Para evitar este problema, en caso de que el uso de memoria por parte del CPU supere la cantidad de memoria física, la figura se invertirá mostrando, en la parte superior, el indicador de memoria y por debajo, el indicador de uso de la CPU.

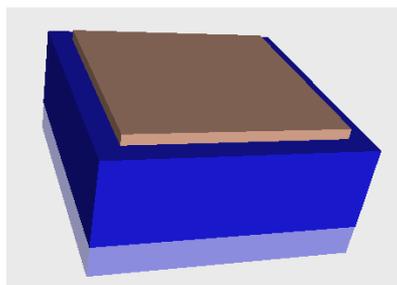


Figura 8. La representación de un nodo cuya CPU se encuentra utilizando memoria virtual

## 4.2. Parámetros a nivel de cada uno de los procesos en un nodo

Los prismas rectangulares no son los únicos elementos con los que contará la visualización. Hasta el momento, los indicadores visualizados corresponden en su totalidad a parámetros generales del nodo. Existen otros indicadores que pueden ser visualizados para un nodo y que sin duda el usuario deseará en cierto momento poder visualizar. Sin embargo, mostrar toda la información posible de todos los nodos generaría un volumen de datos que sería imposible de manejar perceptualmente por el usuario. Una técnica utilizada tanto en Visualización de Información como en Interacción Humano-Computadora es el uso de Zoom Semántico. Mediante el Zoom Semántico la representación ampliada de un objeto no es simplemente un escalado de su aspecto geométrico sino que se presenta información adicional que, de ser necesario, implicará un cambio de representación del mismo. Definiremos tres niveles de escalado o zoom: cada nivel ofrece más información que el anterior. En el primer nivel contaremos con la identificación del nodo, una cadena de caracteres que se situará por sobre la representación gráfica en todo momento. Cuando el usuario sitúe el puntero del mouse sobre un nodo, automáticamente se podrá ver junto a éste un resumen de la información relevante y al realizar un click se podrá observar el listado de los procesos que se están ejecutando junto con sus parámetros asociados. Para esto el prisma asociado a la cantidad de memoria aumentará su altura y la información sobre los procesos será visualizada sobre dos de sus caras.

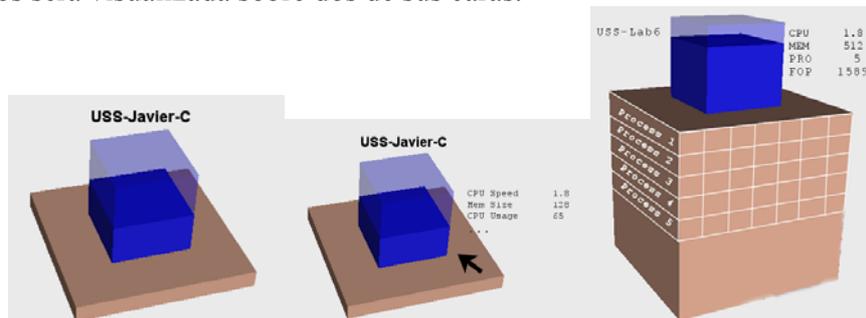


Figura 9. Los tres niveles de zoom semántico definidos.

## 4.3. Topología del sistema distribuido

Un parámetro que no pertenece al nodo ni a los procesos en el nodo es la topología de la red. En este caso, la topología de la red se puede visualizar al ubicar los nodos conectados de forma tal que en conjunto se imite la forma de la red física.

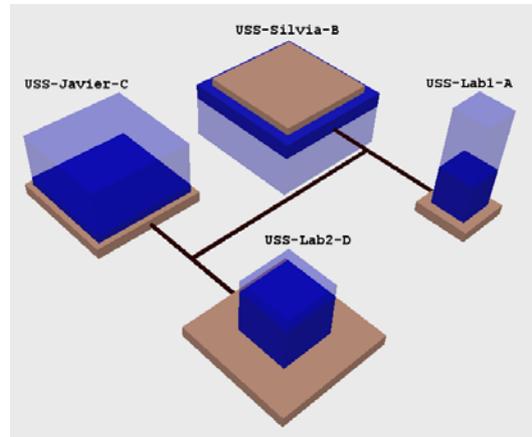


Figura 10. La vista del cluster de computadoras.

La red de nodos que forman el sistema distribuido conforman un grafo y la navegación de esta red es equivalente a la de un grafo. Cuando un usuario solicita ver la información detallada de un nodo se encuentra obligado a visualizar sólo el nodo seleccionado y no el resto de la red; esto es debido a que no se pueden ajustar ambos elementos al tamaño de un display. En estos casos, el usuario no cuenta con una referencia visual de lo que ocurre en el resto del sistema por lo que se pierde la información de lo que sucede en las partes no visualizadas. Es necesario contar con una técnica que nos permita integrar el elemento sobre el cual se centra la atención del usuario y su contexto. Desde la Visualización de Información y la Interacción Humano-Computadora se plantearon soluciones a problemas análogos en diferente contexto. Una de las soluciones utiliza una distorsión de la vista natural de la estructura mediante la aplicación de transformaciones matemáticas a la representación visual. Esto permite distinguir dos áreas: por un lado un área central o foco que es magnificada y por el otro, el resto del contexto que se presenta comprimido espacialmente. Las técnicas que implementan esta solución logran mapear estructuras completas dentro de la pantalla y proveen información tanto local como global; esta técnica se denomina *FishEye View*.

#### 4.3.1. FishEye View

La técnica de *FishEye View* es única en el sentido que utiliza un sistema de prioridades para determinar qué información debe ser presentada y cuál debe ser suprimida. La aplicación de esta técnica en una dimensión es similar a Pared de Perspectiva [2]; sin embargo, la discontinuidad que se generaba en este último caso entre el foco y el contexto es reemplazada con una transición exponencial.

Un *FishEye View* de un grafo muestra un área de interés con gran detalle y aumentado de tamaño, y el resto de las áreas con un menor tamaño y una disminución del detalle en forma progresiva a medida que la distancia de los elementos al área de interés aumenta. Esta integración entre detalle local y contexto global se logra modificando la posición y el tamaño de cada elemento.

Dado el grafo de la figura 11a como base, se ve en la figura 11b el resultado de aplicar un *FishEye View* sobre uno de sus nodos. En esta vista, el vértice con el borde grueso corresponde al área de interés actual: este punto es llamado **foco**. El tamaño y el detalle de cada nodo en el *FishEye View* dependen de la distancia entre el foco y el nodo, el valor de importancia asociado al nodo y otros parámetros.

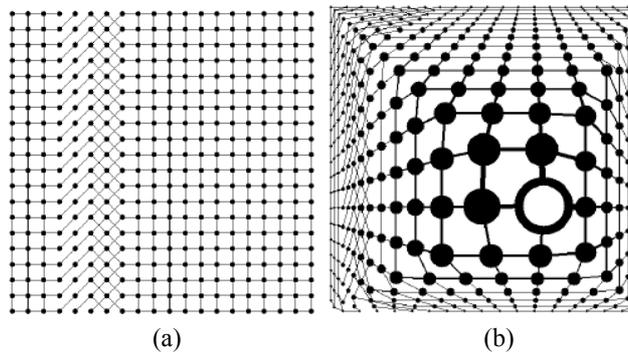


Figura 11. Un grafo (a) y el resultado de aplicar FishEye View sobre el mismo (b),

La vista del grafo de la figura 11a se conoce como **vista normal** y sus coordenadas, como **coordenadas normales**. Cada vértice posee una posición especificada por su coordenada normal y un tamaño y también un valor numérico que representa su importancia en la estructura global, este número se identifica como **a priori importante, API**.

Las coordenadas del grafo en el *FishEye View* son las **coordenadas FishEye**. Cada vértice en el *FishEye View* se define por su posición, su tamaño y la cantidad de detalle con el que se muestra. Finalmente, cada vértice posee un **valor visual, VV**, calculado en base a su distancia al foco en coordenadas normales y su API.

Generar un *Fisheye View* implica magnificar aquellos vértices de mayor interés y desmagnificar aquéllos de menor interés. Además, la posición de todos los vértices debe ser recalculada para poder obtener más espacio para la sección magnificada, de modo tal que toda la vista ocupe el mismo espacio en pantalla. La *posición* de un vértice en el *Fisheye View* depende de su posición en la vista normal y de su distancia al foco. El *tamaño* del vértice en el *Fisheye View* depende de su distancia al foco, su tamaño en la vista normal y su API. La *cantidad de detalle* con el que se muestra un nodo estará en función de su tamaño en el *Fisheye View*. Estos conceptos se pueden encontrar formalizados en [3]. La aplicación de la vista de pez sobre un grafo consta entonces de dos tareas:

- Aplicar una transformación geométrica a la vista normal para reubicar los vértices, lo que permite magnificar y desmagnificar las áreas cercanas y lejanas al foco respectivamente.
- Utilizar el API de cada vértice para obtener su tamaño, detalle y peso visual correspondiente.

Al combinar estas tareas se obtiene una nueva vista que prioriza el elemento en foco pero no pierde su contexto. *Fisheye View* es una técnica que presenta grandes ventajas en su utilización para la visualización de grandes estructuras. Es importante remarcar que ésta no es la única técnica existente para este fin y se deja pendiente para futuros trabajos el análisis y desarrollo de nuevas técnicas.

### 4.3.2. Visualización del Balance de Carga con Fisheye View

En la figura 12 se puede ver la vista normal de un sistema distribuido y en la figura 13 vemos un FishEye View aplicado sobre el nodo USS-Lab6. No solo se magnifico el foco para mostrar mas información sino que se demagnificaron los nodos restantes para lograr incluir todos los elementos en una misma vista.

## 5. Conclusiones y trabajo futuro

El resultado de combinar los elementos visuales antes descritos en una visualización para determinar la carga en un sistema distribuido nos da como resultado una vista intuitiva del estado actual de nuestro sistema.

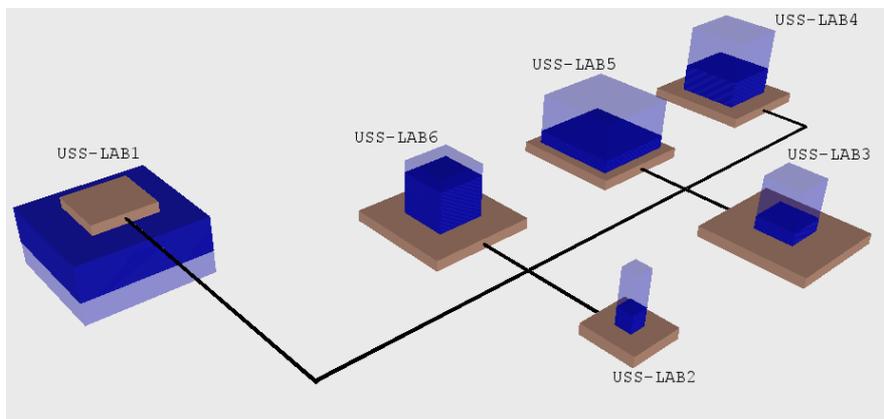


Figura 12. Un sistema distribuido con 6 nodos.

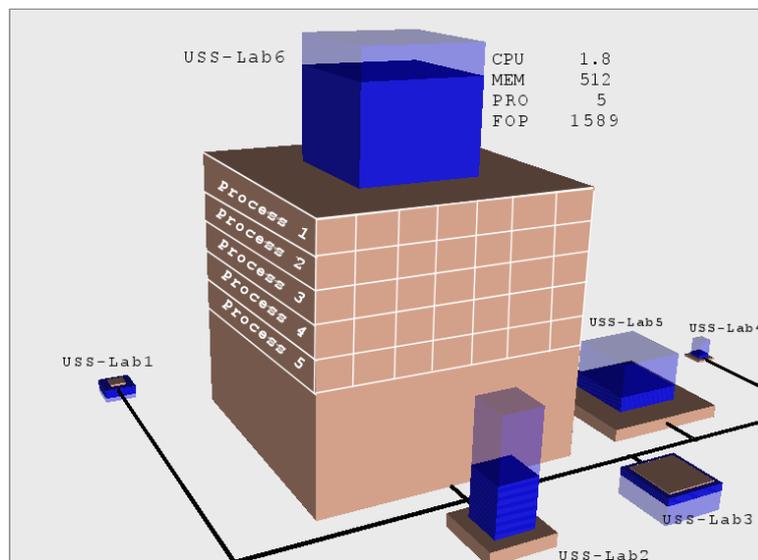


Figura 13. La vista de los procesos en el nodo USS-Lab6 simulando la técnica de FishEye View.

En esta etapa inicial de investigación los usuarios de la visualización propuesta son los investigadores y desarrolladores del LISiDi. Desde el punto de vista de la visualización estamos trabajando en una extensión para que no sólo sea una vista del estado sino que se constituya en una interfaz de control del sistema. Se le desea permitir al usuario interactuar con cada nodo brindando la posibilidad de trabajar sobre el nodo en forma remota. Se está analizando cómo el usuario interactuará con la visualización para poder migrar procesos y de qué forma la visualización representará dicha migración. Otro objetivo a futuro es la posibilidad de brindar más de un foco sobre la vista. Al permitir esto, el usuario podrá tener el máximo detalle de información de dos o más nodos permitiendo una sencilla y eficiente comparación. Finalmente se está evaluando la posibilidad de incluir la visualización de procesadores SMP, *Symmetric Multiprocessing*.

## 6. Bibliografía

- [1] Sergio Martig, Guillermo Trutner, Mercedes Vitturini, Carlos Alvez, Sandra Di Luca, Sivia Castro, Javier Echaiz, Jorge Ardenghi. “*Visualización del Balance de Carga en un Sistema Distribuido*”. Jornadas Chilenas de Computación. 2001.
- [2] Gerstmann, Derek. “*Advanced visual interfaces for hierarchical structures*”. Human Computer Interaction. CSS480, Research Paper, Human Computer Interaction. March 8, 2001.
- [3] Sarkar, Manojit and Brown, H., Marc. “*Graphical FishEye views*”. Communication of the ACM. December 1994/Vol 37, No. 12.
- [4] Robertson, G., Card, S., Mackinlay, J., “*Information Visualization Using 3D Interactive Animation*”, Communications of the ACM, 36(4), pp. 56-71, 1993.
- [5] Card, S., Mackinlay, J., Shneiderman, B., “*Readings in Information Visualization – Using Vision to Think*”, Morgan Kaufmann, 1999.