Propiedades Dinámicas de Clusters de Computadoras para Cómputo Distribuido

Magali Arellano Vázquez† y Miguel Robles Pérez

†Facultad de Ciencias, UAEMor, Cuernavaca, Morelos. CP 62209
CIE-UNAM, Temixco, Morelos. CP 62580
{avm,mrp}@cie.unam.mx

Resumen: El propósito de este trabajo es estudiar la dinámica de migración de los programas que se ejecutan en un cluster, para proponer nuevos métodos que midan la influencia de variables aleatorias (como la participación de múltiples usuarios, la ejecución de procesos de sistema o la heterogeneidad del hardware del cluster en la ejecución de programas). Se busca que estos métodos estén basados en la metodología de la física estadística y se busca establecer analogías entre el movimiento de partículas en un baño térmico (movimiento Browniano) y la migración de procesos. Finalmente, se pretende encontrar una medida global que proporcione información del comportamiento del cluster. Se introduce como método alternativo la definición y el análisis de grafos generados de manera dinámica al mover uno o múltiples procesos por el cluster.

Palabras clave: Cluster heterogéneo, movimiento Browniano, migración de procesos, grafos aleatorios

1 Introducción

Un cluster de computadoras es un grupo de computadoras débilmente acopladas que trabajan juntas, comparten recursos de hardware y se comunican por una red de alta velocidad, pero en muchos aspectos se comportan como si fuesen una única computadora más potente que las comunes de escritorio [1].

El desempeño de los clusters es relativo; regularmente se sabe con qué recursos se cuenta sumando las capacidades de almacenamiento y procesamiento de los componentes del cluster, pero en la realidad, no se tienen estos recursos de manera íntegra; ya que en todos los sistemas operativos siempre hay procesos ejecutándose sin que el usuario tenga conciencia de esto y en el caso de los sistemas distribuidos, existen más procesos ejecutándose provenientes de otros usuarios y de los programas que éstos utilizan [4].

El objetivo de este trabajo es estudiar y caracterizar las propiedades dinámicas de procesos como: migración, trayectorias, permanencia, probabilidad de permanencia o movimiento dentro del cluster en distintas condiciones.

M.A Cruz-Chávez, J.C. Zavala-Díaz (Eds): CICos 2008, ISBN:978-607-00-0165-9. pp. 100-111, 2008

La meta principal es buscar nuevos métodos para medir la influencia de variables aleatorias como la participación de múltiples usuarios, la ejecución de programas del sistema o la heterogeneidad del hardware, todo esto en el funcionamiento del cluster.

Se buscan métodos que estén basados en la metodología de la física estadística, estableciendo una analogía entre la migración de procesos, el movimiento de una partícula en un baño térmico (conocido como movimiento Browniano) [12] y el proceso de evolución de grafos aleatorios, para encontrar medidas globales que nos permitan conocer el comportamiento del cluster en que habitan.

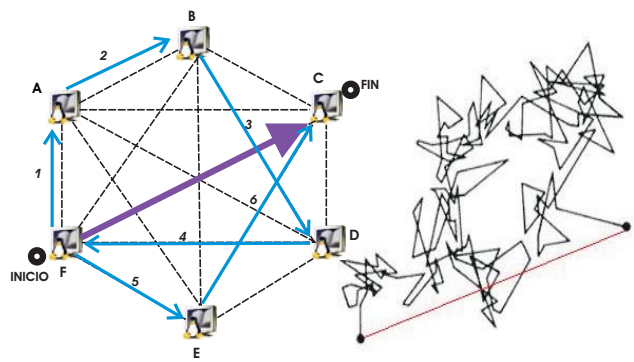

Fig. 1. Migración de procesos en una red (izquierda). Trayectoria de una partícula con movimiento browniano (derecha)

El movimiento Browniano es el movimiento que siguen partículas pequeñas suspendidas en un fluido y se manifiesta como un movimiento en trayectorias aparentemente al azar. En realidad, los átomos del líquido colisionan con el cuerpo suspendido causando un movimiento complejo y por tanto aleatorio [5]. Como los átomos en el líquido están también distribuidos al azar, las colisiones entre ellos y un cuerpo suspendido en el líquido sucederán impredeciblemente causando el efecto observado. En el movimiento Browniano se distingue la distancia recorrida del desplazamiento (Fig. 1).

- Cuando nos referimos a la distancia recorrida, se hace referencia a la longitud de toda la ruta recorrida
- En el caso del desplazamiento, se hace referencia solo a la diferencia entre el punto de inicio y el de destino.

La migración de un proceso puede tener analogías con el movimiento Browniano (Fig. 1) pues el camino que recorre un programa está determinado por una multitud de factores que pueden considerarse aleatorios que son muy difíciles de predecir de manera determinista pero tratables probabilísticamente. Así mismo ambos fenómenos son similares al movimiento de un caminante al azar [2] [6], el cual es la formalización de la idea intuitiva de una sucesión de pasos (cada uno en una dirección aleatoria). Una partícula suspendida en un líquido puede modelarse como un caminante al azar en 3 dimensiones con pasos continuos. En el caso de una dimensión, el caminante corresponde a una partícula moviéndose aleatoriamente en el conjunto Z de enteros. Iniciando en el origen al tiempo 0 y en cada tiempo $1,2,\ldots,n$, se mueve un paso a la derecha con probabilidad p, o un paso a la izquierda con

probabilidad *p-1*, tal que $0 < p < 1$. El problema es determinar la probabilidad de que el caminante se encuentre en una posición *x* en un tiempo *t*.

El problema al que nos enfrentamos en el caso del cluster puede verse como un caminante al azar pero en una red (Fig. 1) lo que le añade complejidad al problema y otro factor importante de considerar: la probabilidad de que el caminante permanezca o no en un nodo.

Cuando un proceso migra de un nodo a otro, éste describe una trayectoria que puede ser considerada como un grafo que depende directamente de las condiciones de operación y de la topología del cluster, entonces este grafo es un grafo aleatorio[1]. Las redes con topologías muy complejas y con principios de organización desconocidos pueden parecer ser grafos aleatorios [3]. La teoría de los grafos aleatorios fue introducida por Paul Erdös y Alfred Rênyi. Ellos introdujeron lo que se llama: modelo Erdös-Rênyi [9].

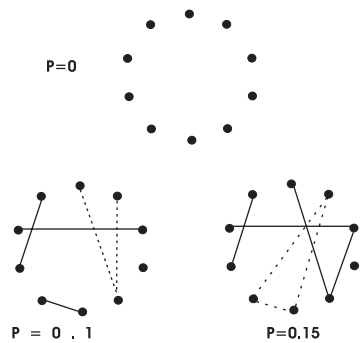

Fig. 2. El proceso de evolución de grafos por el modelo Erdos-Rényi. Esta gráfica muestra 2 diferentes estados del desarrollo de los grafos, correspondientes a *p=0.1* y *p=0.15*.

La construcción de grafos aleatorios es llamada: evolución. Comenzando con un conjunto de *N* vértices aislados, el grafo se desarrolla agregando sucesivamente aristas aleatorias (ver Fig. 2). Los grafos obtenidos en las diferentes etapas de este proceso corresponden a la mayor probabilidad de conexión *p*, eventualmente obtendremos un grafo totalmente conexo para *p=1* [9].

Es claro que un programa ejecutándose en un cluster es similar a una partícula en movimiento Browniano y al moverse dibuja trayectorias iguales a los grafos que pueden generarse dentro de la red formada por las unidades del cluster. Este es el punto de partida del trabajo, la descripción de las herramientas y las propuestas de estudios se discuten en las secciones siguientes.

[1] Un grafo aleatorio es aquel en el cuál las aristas están distribuidas aleatoriamente

2 Herramientas

Para este estudio se utilizó un prototipo de cluster heterogéneo utilizando un conjunto de computadoras personales y el software ClusterKnoppix [17] que implementa las características del software OpenMosix [14] en un LiveCd, también se utilizó el software de virtualización QEMU [16], estas herramientas se describen a continuación.

2.1 OpenMosix

OpenMosix es una bifurcación del proyecto MOSIX [7]. Entre las principales características de sistema de imagen única implementadas, esta el equilibrio de carga computacional entre los nodos de manera dinámica y permite al usuario un espacio para el acceso a archivos similar a NFS[2] pero distribuido entre los nodos del sistema [8]. OpenMosix aporta la funcionalidad básica de cluster middleware. Un parche al núcleo de sistema es el responsable de convertir un conjunto de computadoras interconectadas a través de una LAN, en una supercomputadora, permitiendo que los nodos trabajen en estrecha cooperación.

Los algoritmos que se encargan de compartir recursos (Resource Sharing Algorithms) del sistema OpenMosix están diseñados para responder en tiempo real a las variaciones del uso de los recursos entre los nodos que componen el cluster [11]. Esto se logra a través de la migración de procesos de un nodo a otro de manera transparente para el usuario, con el objeto de equilibrar la carga entre los nodos del cluster. Así el objetivo es mejorar el funcionamiento del sistema complejo y crear un ambiente multiusuario de tiempo compartido para la ejecución de aplicaciones secuénciales y paralelas.

La distribución de GNU/Linux Knoppix, basada en Debian y que utiliza KDE [15]; Knoppix es un LiveCd[3], por tanto, no requiere una instalación en el disco duro; reconoce automáticamente la mayoría del hardware del ordenador soportado por Linux cuando se inicia. Así mismo provee las mismas características de Linux, más las funcionalidades de OpenMosix [10].

La principal debilidad de OpenMosix radica en la dificultad de usarlo en núcleos modernos de Linux (versión 2.6) y en ambientes nativos de 64 bits, sin embargo, usando el concepto de virtualización es posible crear ambientes de prueba que permitan realizar experimentos. Virtualizar significa generar vía software una computadora dentro de otra, de manera que simule un ambiente completamente separado y por tanto la ejecución de sistemas operativos completos. En la actualidad existen virtualizadores de código abierto y en particular elegimos el QEMU.

El programa QEMU es un emulador de procesadores que se basa en la dinámica de traducción binaria; usa una técnica de traducción dinámica que consiste en convertir el código binario de la arquitectura fuente en código entendible por la arquitectura huésped. QEMU también tiene capacidades de virtualización dentro de un sistema

[2] Network File System

[3] Es un CD que contiene todas la funcionalidades de un sistema operativo completo sin tener que instalarse en el disco duro

operativo, ya sea Linux, Windows, o cualquiera de los sistemas operativos admitidos (de hecho es la forma más común de uso). Esta máquina virtual puede ejecutarse en cualquier tipo de Microprocesador o arquitectura (x86, x86-64, PowerPC, MIPS, SPARC, etc.). Está licenciado en parte con la LGPL y la GPL de GNU [16].

Mediante las herramientas descritas se implementaron arreglos de computadoras distribuidas en ambiente real y virtual para su uso experimental. A continuación describimos los métodos y resultados obtenidos.

3 Metodología

El estudio se dividió en 2 etapas:

1. La primera es la implementación del cluster. Para ello se decidió utilizar un grupo computadoras de personales que operan como estaciones de trabajo Linux, utilizando el LiveCd de ClusterKnoppix, para armar un cluster heterogéneo, con un número de máquinas seleccionadas ad-hoc para el experimento que se decidió plantear.
2. En la segunda etapa consistió en llevar a cabo pruebas con configuraciones diferentes, específicamente se decidió comenzar con 2 y 3 máquinas, sin presencia de usuarios externos y con procesos controlados. Estos procesos son monitoreados con la aplicación OpenMosixView para seguir su comportamiento en el cluster. Para comparar el caso de ambiente controlado con el del ambiente real, se efectuarán pruebas con un cluster virtual.

En analogía a los conceptos propios de movimiento Browniano se introdujeron las siguientes definiciones con el fin de establecer entidades medibles.

- **Trayectorias:** Se define ``trayectoria" de un proceso como el camino que desarrolla un proceso al migrar entre los nodos del cluster en un tiempo dado.
- **Permanencia:** Se define ``permanencia" como la cantidad de segundos que un proceso se ejecuta en un nodo; por lo tanto la ``distribución de permanencia de los procesos", se define como el número de procesos que se ejecutaron en un nodo durante un intervalo de tiempo dado.
- **Cambio de procesador:** Se define ``cambio de procesador" cuando un proceso deja de ejecutarse en el nodo local y migra a otro; por lo tanto, la distribución de cambio de procesador son las veces que un proceso migra de un procesador a otro en un intervalo de tiempo dado.
- **Velocidad:** Se define ``velocidad"como el número de cambios de un proceso a través de los nodos del cluster en un intervalo de tiempo dado.

Los grafos aparecen de forma natural durante el funcionamiento del cluster, pues cada trayectoria dibuja uno. Sin embargo aquí proponemos una definición de grafo relativa más al estado del cluster que a la propia trayectoria. Así asociaremos al grafo la información del número total de nodos del cluster, el nodo donde se originaron los procesos y la manera en que estos pueden migrar. En Fig. 3 podemos observar un ejemplo de grafos para 1 y 3 procesos, con 3 nodos, la estrella etiqueta al nodo generador y las flechas indican cuantos procesos migraron al nodo que apuntan.

Para calcular el número de grafos posibles con 3 procesos consideremos que tenemos 3 máquinas, m_1, m_2 y m_3, m_1 recibe n procesos, puede quedarse con algunos

(o con ninguno) y distribuye los restantes entre m_2 y m_3 (no de forma equitativa) como se puede apreciar en la Fig. 3.

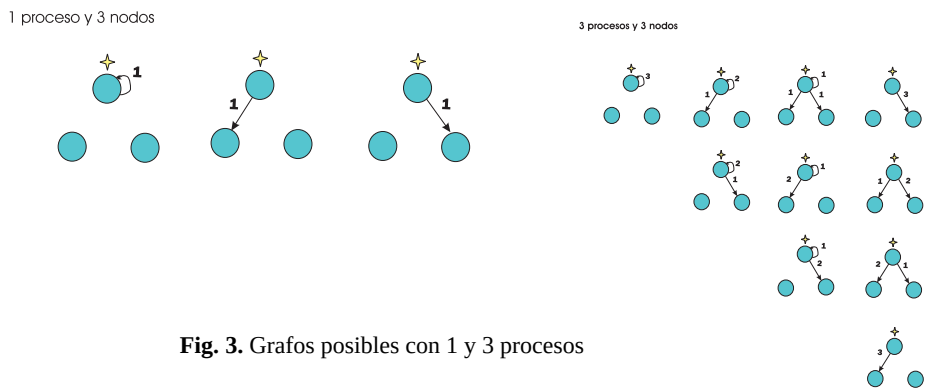

Fig. 3. Grafos posibles con 1 y 3 procesos

Existen 2 casos:

1. No se distingue entre procesos, sólo importa cuántos se mandan a cada máquina ($\sum_{i=1}^{n+1} j$).
2. Se distingue entre procesos e importa a qué máquina se envía ($n!\sum_{i=1}^{n+1} j$).

Se generaron todos los grafos posibles para el caso de 3 nodos y hasta 9 procesos. Se ha asignado un identificador numérico de 3 dígitos. Lo importante aquí es conocer qué grafos son más frecuentes en determinadas circunstancias, así un grafo es un análogo al estado del sistema físico.

Definamos distribución de grafos; se define como la probabilidad de un grafo de estar presente en un intervalo de tiempo dado, cuando hay *n* nodos y *m* procesos.

Para mayor detalle de los experimentos y del tratamiento de la información consultar [13]. Esta estadística es realizada en el sistema propuesto y se presenta junto con los resultados en la siguiente sección.

4 Resultados

Se llevaron a cabo una serie de experimentos, algunos en ambientes controlados y con máquinas dedicadas y otro experimento en un cluster virtual, en esta sección se exponen los resultados obtenidos.

4.1 Experimento en ambiente controlado

Se ocuparon 3 nodos, con las siguientes características:

- **Nodo 0 (atl):** Athlon MP 2600 Dual, 1 GB RAM
- **Nodo 1 (cuauhtli):** Opteron 244 Dual 1800 1 GB RAM
- **Nodo 2 (tlamacazqui):** Athlon 64 x2, 1 GB RAM

No se analizaron procesos de sistema, sino un programa de simulación de dinámica de partículas durante un intervalo de más de *900* segundos, se siguió la trayectoria de 12 procesos que fueron aumentando uno a uno dejando un intervalo de tiempo mayor cuando el número de procesos era impar, obteniendo la Fig. 4.

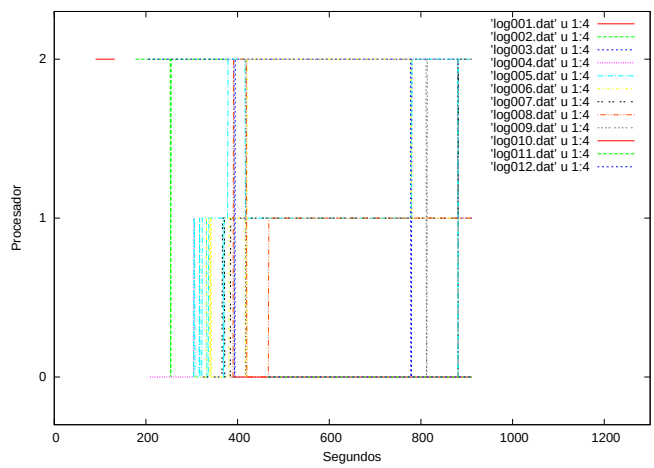

Fig. 4. Distribución de probabilidad de los grafos presentes en el experimento

Distribución de permanencia: En este experimento se cambió la manera de calcular la distribución de permanencia de los procesos, dado que esta vez se toma como marco de referencia los nodos y no el proceso seguido. Se dividió el tiempo total del experimento en intervalos de 50 segundos, por lo que los intervalos quedan de la siguiente manera:

Intervalo	Segundos	Intervalo	Segundos
1	0-50	8	351-400
2	51-100	9	401-450
3	101-150	10	451-500
4	151-200	11	501-550
5	201-250	13	551-600
6	251-300	14	601-650
7	351-400	14	651-700

Las distribuciones obtenidas en los distintos nodos se muestran en Fig. 5.

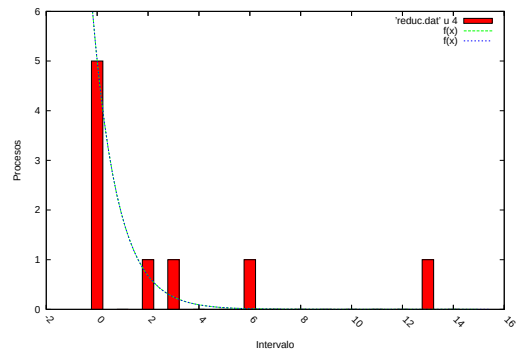

Fig. 5. Cambios de procesador en intervalos de 50 segundos, gráfica ajustada a la función $f(x) = e^{-bx}$ en nodo 3

Velocidad: La distribución de la velocidad a lo largo de este experimento se muestra Fig.6.

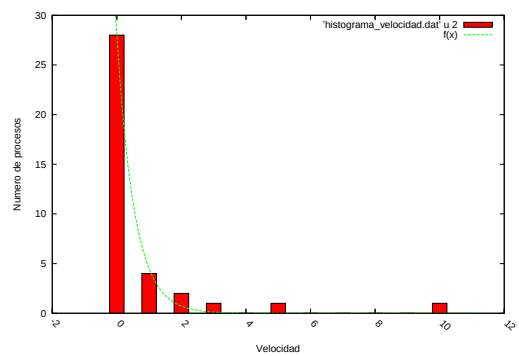

Fig. 6. Distribución de velocidades, distribución ajustada a la función $f(x) = e^{-bx}$

4.2 Distribución de grafos

Se ha analizó la presencia y distribución de los grafos de los 219 posibles a lo largo del experimento, el resultado se muestra en Fig. 7.

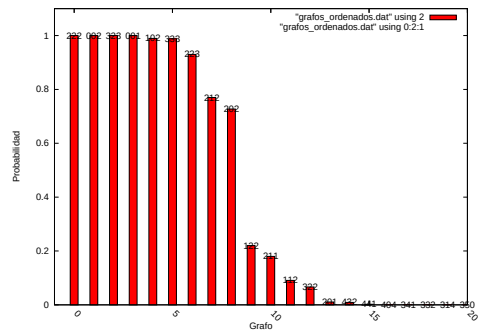

Fig. 7. Distribución de probabilidad de los grafos presentes en el experimento

4.3 Experimento en cluster virtual

En este experimento se utilizaron máquinas virtuales para armar el cluster, utilizando QEMU y se ejecutaron las pruebas con un solo proceso. Se obtuvo la trayectoria (Fig. 8), la velocidad (Fig. 9) y la distribución de grafos (Fig. 10)

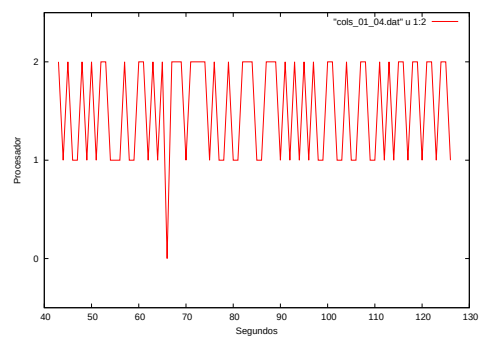

Fig. 8. Trayectoria de un proceso en el experimento en un cluster virtual

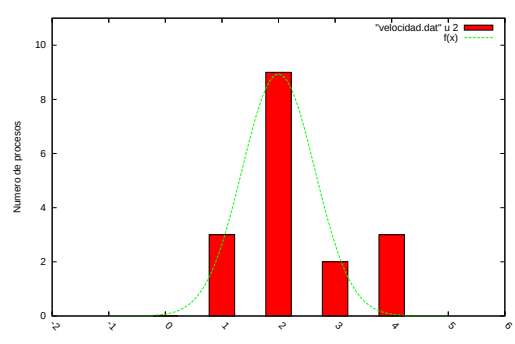

Fig. 9. Velocidad de procesos en la máquina virtual, ajustada a la función $f(x) = ae^{-b(x-2)^2}$

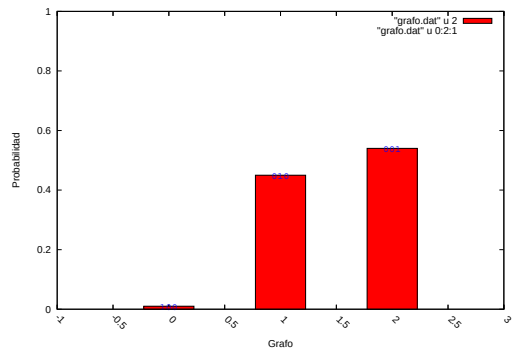

Fig. 10. Distribución de los grafos presentes en la máquina virtual para un proceso

5 Discusión

Experimento en ambiente controlado: El análisis de trayectorias aportó información como:

1. Conocer el rendimiento y capacidad de la máquina local con respecto a los demás nodos.
2. Saber que tan ruidoso es el sistema.
3. El análisis de la velocidad del cluster es útil para saber qué tanto influyen los factores externos del cluster en su rendimiento.
4. La distribución de velocidades en un cluster con máquinas dedicadas se ajusta a la función $f(x) = e^{-bx}$ esta puede ser una condición general en ambientes sin *ruido*.
5. Se detectó una mayor frecuencia de aparición de 12 grafos en particular, de 219 posibles (Fig. 11).

Experimento en cluster virtual: La característica fundamental en el experimento es el haberse realizado en un cluster ``virtual" y por tanto es de esperar que la ejecución de programas en él sea mucho más sensible a las múltiples cargas de los sistemas huésped y cliente. Esta prueba además de información de trayectorias y permanencia, aportó información para analizar un cluster con un ambiente no controlado y la primera diferencia esta en la distribución de velocidades es muy distinta a la reportada para el caso del cluster dedicado, dado que el cluster con ambiente no controlado parece ajustarse a la función $f(x) = ae^{-b(x-2)^2}$ la cual es conocida como una distribución Gaussiana. Este resultado es importante pues mediante una definición para la velocidad de los procesos es posible encontrar una función de distribución similar a la que tienen partículas moviéndose en un baño térmico y por ende el parámetro *b* que en sistemas físicos es proporcional al inverso de la temperatura, podría establecerse como una definición para la ``temperatura

informática" del cluster, que finalmente es la analogía buscada. Esta propuesta es aún preliminar pues para establecerla es necesario contar con más muestras estadísticas en diferentes escenarios.

Finalmente, se puede decir que el experimento utilizando virtualización con QEMU aportó información útil para tener la referencia de un cluster con mucha actividad y así poder comparar los datos con los obtenidos de los experimentos anteriores.

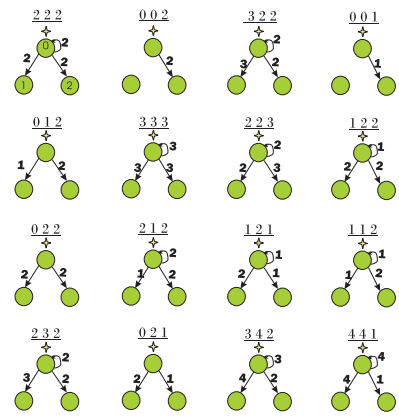

Fig. 11. Grafos más frecuentes durante el experimento

6 Conclusiones

El análisis de los datos estadístico fue útil para conocer el comportamiento del cluster, los resultados obtenidos nos indican que en máquinas dedicadas los procesos llegan pronto al equilibrio, es decir, a situaciones permanentes en las que el sistema considera que el ``costo" es menor; esto correspondería en una analogía física a estados de mínima energía.

Fue posible comparar el comportamiento de un cluster dedicado contra un cluster con influencia de variables aleatorias (como usuarios y procesos de usuario y sistema) y el comportamiento es distinto; en todos los aspectos medidos, en el cluster con variables se presenta mayor migración, por lo tanto trayectorias más alternantes y una distribución de velocidades distinta.

Los grafos presentes en el experimento no son aleatorios porque la distribución que presentaron a lo largo del experimento no genera que aparezcan con diferente probabilidad. Los grafos que tienen mayor probabilidad de estar presentes en el cluster son aquellos que distribuyen la carga casi equitativamente entre los nodos de un cluster dedicado. En el cluster virtual es un tanto distinto, dado que influye la carga de cada procesador en particular porque los procesos tienden a migrar hacia el nodo que tiene más recursos disponibles.

La conclusión más importante del trabajo es el trazar una metodología que permite proponer índices para la medición del desempeño del cluster, en particular la analogía

que permite proponer la temperatura informática es destacable. Sin embargo, los resultados obtenidos no son concluyentes, puesto que los recursos fueron limitados, se trabajó con pocos datos y los experimentos no fueron repetidos las suficientes veces para tener una estadística confiable.

Referencias

1. Tanenbaum, Andrew S y Van Steen,Maarten: Distributed Systems, principles and paradigms. Universiteit. Amsterdam (2002)
2. Kenneth H. Rosen: Handbook of discrete and combinatorial mathematics. AT&T Laboratories (2000)
3. Andrzej Rucinski, Svante Janson, Luczak Tomas: Random Graphs. Adam Mickiewicz University (2000)
4. Byya,Rajkumar: High performance cluster computing.(1999)
5. M. P. Allen, D. J.Tildesley: Computer simulation of liquids. Oxford, UK (1989)
6. Spitzer, Frank: Principles of random walk. Graduate Texts in Mathematics. Springer (2001)
7. Joseph D. Sloan: High performance linux clusters with Oscar, openMosix and MPI. O'Reilly (2004)
8. Freddy J. Perozo Rondón: CLUSTER MANGOSTA: Implementación y evaluación. Faraute de Ciencias y tecnología. Julio (2006)
9. Réka Albert, Albert-László Barabási: Statistical mechanics of complex networks. Reviews of Modern Physics. January (2002)
10. Ian Latter: HowTo Heterogeneous Clusters; Running ClusterKnoppix as a master node to a CHAOS drone army. Macquarie University, Australia, (2003)
11. Moshe Bar, Maya, Asmita, Snehal,Krushna: openMosix. Linux Congress 2003. http://openmosix.sourceforge.net/linux-kongress_2003_openMosix.pdf (2003)
12. Einstein Albert: On the motion of small particles suspended in liquids at rest requiered by the molecular kinetic theory of heat. Annalen der Physik (1905)
13. Arellano Vázquez Magali: Propiedades dinámicas de cúmulos de computadoras para cómputo distribuido. Tesis de Licenciatura. UAEMor (2008)
14. OpenMosix Project http://openmosix.sourceforge.net/
15. Knoppix, GNU/Linux distribution , http://www.knoppix.net/
16. QEMU, Open Source Processor Emulator, http://fabrice.bellard.free.fr/qemu/
17. Clusterknoppix project, http://clusterknoppix.sw.be/