Diseño e implementación de un simulador para explorar la cooperación en entornos distribuidos

Davide Vega, Roc Messeguer, Felix Freitag

{dvega,messeguer,felix}ac.upc.edu
Universitat Politècnica de Catalunya
Departamento de Arquitectura de Computadores
Campus Nord Edificios D6/C6, 08034 Barcelona, España

Abstract. Los avances tecnológicos en los sistemas de comunicación y computación móviles están impulsando la emergencia de sistemas colaborativos distribuidos. No obstante, la participación de los usuarios se ve frenada por los escasos recursos disponibles en los dispositivos móviles actuales. En este artículo se presenta un simulador de redes colaborativas descentralizadas que permite estudiar cómo la topología de la red y las estrategias de los diferentes participantes pueden incentivar y mejorar su participación activa –colaboración– en este tipo de escenarios. Los resultados obtenidos en las simulaciones realizadas sobre redes de overlay, nos han permitido realizar observaciones importantes en cuánto a cómo distribuir los recursos en las redes con dispositivos heterogéneos, qué tipo de estrategias de colaboración y colocación de nodos o qué topologías pueden incentivar la participación activa. Por ello, consideramos el simulador una buena herramienta para analizar este tipo de escenarios desde un punto de vista empírico.

1 Introducción

El aumento de potencia de cálculo que requieren las aplicaciones informáticas actuales, las cuales cada vez utilizan más recursos; en cantidad (memoria RAM, espacio de disco duro, etc.) y calidad (altas frecuencias de CPU, baja latencia de las tarjetas de red, etc.) ha dejado de manifiesto una importante falta de recursos computacionales en los ordenadores personales. Como consecuencia, en muchos casos los pocos recursos físicos disponibles se transforman en una limitación temporal, que no es suficiente para justificar su actualización pero que sí impide el desarrollo normal del flujo de trabajo del usuario o la ejecución de tareas complejas, aunque estas sean esporádicas.

En contraposición, los avances tecnológicos que la industria electrónica ha conseguido estos últimos años también están propulsando un significativo incremento de dispositivos de pequeño tamaño con muchas prestaciones equiparables a la de los ordenadores de escritorio, y otras nuevas todavía en vía de exploración, como la capacidad de movilidad o de geolocalización. En este nuevo escenario, parece lógico plantear soluciones híbridas que permitan a los dispositivos móviles (*tablets, smartphones*, etc.) aprovechar su capacidad de ubicuidad y conectividad para utilizar los recursos que le proporciona su entorno con el fin de crear redes

colaborativas entre dispositivos móviles y de escritorio, en las que los participantes puedan adquirir temporalmente parte de los recursos de otro dispositivo durante un tiempo determinado para realizar una tarea o parte de ella.

Uno de los retos más importantes a los que se enfrentan las arquitecturas distribuidas de computación con dispositivos móviles es la gestión óptima y justa de sus recursos [12] [14]. Debido a su escasez, un usuario puede decidir realizar o no realizar una determinada tarea en función de la carga de trabajo actual de su dispositivo o, de forma conservadora, en función de la cantidad de energía que le quede para finalizar su jornada. Por lo tanto, al usuario se le debe proporcionar una retribución justa por sus recursos, que le motive a compartirlos. No obstante, siempre existirán usuarios egoístas que intentarán sacar mayor beneficio personal de estos sistemas sin comprometer sus propios recursos. Si esto ocurre, se creará una sensación de rechazo en aquellos usuarios que resulten más perjudicados y son menos altruistas; frenando así la adopción del sistema.

Diferentes autores han utilizado métodos de incentivos basados en créditos, sistemas de mercado o protocolos de comunicación para asegurar un trato justo de los usuarios en aplicaciones basadas en redes p2p como la que se plantea. No obstante, estas soluciones requieren implementaciones específicas en el *software* de los clientes para poderse llevar a cabo, lo que facilita a los *free riders* encontrar caminos alternativos para evitarlas. Una solución basada en la topología de la red *overlay* como método de incentivo permitiría a los administradores de los clusters y redes distribuidas de computación crear una plataforma segura independientemente del comportamiento de los usuarios de la misma o controlar los diferentes niveles de calidad de servicio que se quiere proporcionar a cada participante. El mismo estudio serviría también para optimizar en coste la infraestructura necesaria para soportar estos servicios.

A continuación se describe un simulador que permite el estudio del comportamiento de redes colaborativas de computación desde una perspectiva topológica de la red, tanto a nivel global como local (nodo a nodo), cuando los dispositivos que la integran participan en una versión del juego del dilema del prisionero. La mayor contribución en este simulador es la fácil implementación de protocolos de negociación dentro del mismo, de forma que su adaptación para dispositivos reales sea inmediata. Parte de esta implementación ha sido descrita anteriormente en [2].

El siguiente capítulo se centra en describir el modelo de sistema utilizado, haciendo énfasis en aquellos aspectos que contribuyen a crear un escenario de prueba para la implementación de algoritmos de negociación. El capítulo 3 describe la arquitectura del simulador y sus componentes más importantes, mientras que en el capítulo 4 se describen algunas de las métricas y resultados que nos permite extraer el simulador. Seguidamente, en el capítulo 5 se muestran algunos resultados extraídos tras el análisis de diferentes topologías a modo de ejemplo de las posibilidades de la aplicación. Finalmente, se enumeran algunos trabajos relacionados y las conclusiones del artículo.

2 Modelado del sistema

Primeramente se exponen algunas suposiciones realizadas en el desarrollo del simulador y la descripción del modelado de los componentes físicos del sistema y el intercambio de información a través de mensajes simples. Seguidamente se hace un breve repaso al *framework* de evaluación de comportamiento, comentando algunos detalles a tener en cuenta en dos de las estrtegias implementadas que pueden adoptar los dispositivos del simulador.

2.1 Dispositivos y recursos

El patrón según el cual podemos modelar la utilización de recursos por parte de un usuario puede variar significativamente dependiendo del tipo concreto de recursos disponibles para compartir. En [3] se definen básicamente dos tipos de recursos: aquellos basados en espacio (discos duros, memoria RAM, etc.) y aquellos basados en tasa (red, ciclos de CPU). Mientras que en el primer caso la petición de una cantidad de recursos r es atómica, los tipos de recursos basados en tasa conservan la propiedad asociativa, por lo que las peticiones pueden adaptarse, agrupándolas o dividiéndolas, en función de la tasa de éxito logrado por las otras peticiones realizadas simultáneamente. Intentar generar un sistema, cuyo modelo tenga en cuenta todas las posibles combinaciones de recursos en la definición de los dispositivos y la gestión de peticiones no tan solo generaría modelos matemáticos complejos, sino que incrementaría la dificultad a la hora de interpretar de los resultados.

Es por ello que en nuestro simulador se ha optado por modelar los dispositivos como nodos de un grafo con una cantidad finita de *slots* o ciclos de CPU cuyo estado para un determinado tiempo t puede ser libre u ocupado; es decir, como un único tipo de recurso basado en tasa. La relación entre la cantidad de recursos que tienen los nodos define su naturaleza, manteniendo una proporción equivalente a la frecuencia de las CPU actuales, que es de 1:5. Por tanto, los dispositivos *handhelds* son nodos que contienen de 1 a 3 slots de CPU, mientras que los ordenadores personales se han modelado como nodos cuya CPU varía entre 14 y 16 slots.

2.2 Peticiones

Aunque no era estrictamente necesario, la implementación de un sistema de información basado en mensajes petición-respuesta basado en [9] [10] ha permitido implementar algoritmos de decisión que luego podrán ser reaprovechados en aplicaciones reales realizando una cantidad mínima de cambios. No obstante, este modelo supone el resto de parámetros de red (comportamiento del canal y/o protocolos de comunicación) ideales, lo que evita interferencias en la lógica del algoritmo de toma de decisiones. Por el contrario, hace necesario adaptar los algoritmos testeados para tener en cuenta estos factores.

Las peticiones se han modelado de forma que un ordenador de escritorio siempre pueda asumir una tarea en su totalidad si así lo desea, pero que requiera

de recursos externos si quiere realizar dos o más tareas. Para incentivar el uso de estos recursos externos también se ha estimado la capacidad de computación de un desktop equivalente a la de tres dispositivos móviles. Estas condiciones mínimas y máximas nos llevan a definir una tarea como una acción que requerirá entre 1 y 10 slots de CPU durante un tiempo t, que variará entre 1 ciclo y 3. Los valores de este último parámetro no influyen en el comportamiento de los nodos, pero los estudios de Vega [2] han demostrado que un valor suficientemente pequeño permite reducir considerablemente el número de ciclos de simulación para tener resultados estadísticamente fiables.

Cada uno de los mensajes, sea de petición o de respuesta, contiene cinco campos: {*source, destination, amount, time, tag*}.

- **source y destination.** Para cada secuencia de negociación entre dos nodos, los campos de source y destination siempre hacen referencia al generador y receptor originales de la petición.
- **amount y time.** Estos dos campos indican la cantidad de slots requeridos y el tiempo durante el cual se reservarán.
- **tag.** Este es un campo de texto libre que otros protocolos de negociación podrían utilizar, pero que no se ha usado en nuestro algoritmo.

Es importante notar que el significado de los campos amount y time varía en función del juego utilizado. En la implementación realizada del dilema del prisionero, por ejemplo, estos campos indican algo diferente en función de cuándo es enviado el mensaje. Esto es, a) la cantidad de recursos requeridos por un nodo n, b) la cantidad de recursos que el nodo m está dispuesto a ceder a n y c) la cantidad de recursos a los que el nodo n renuncia, en caso de exceso.

2.3 Dilema del prisionero

Aunque el simulador permite fácilmente la implantación de cualquier juego cooperativo, a través de su interfaz *Game* se ha incluido en el núcleo una implementación propia del dilema del prisionero. El PD, o dilema del prisionero [1] [4] [5] es un framework ampliamente utilizado para el estudio de estrategias de cooperación entre nodos anónimos, es decir, nodos que no tienen otras alternativas de comunicación entre ellos. En este juego, dos participantes escogen cada turno entre Cooperar (C) o no hacerlo (D). Después de cada ronda, se les informa del resultado de la ronda y se juega la siguiente. La matriz de ganancias y pérdidas para ambas acciones se puede observar en la Tabla 1:

Decisión del jugador	Rival coopera	Rival no-coopera
Cooperar	b-c	c
No cooperar	b	ε

Table 1. Matriz de ganancias y pérdidas del dilema del prisionero

Los diferentes coeficientes de la matriz de ganancias y pérdidas siguen la siguiente regla $b > c > \varepsilon \to 0$, por lo que sitúa a los participantes en el dilema: la cooperación siempre asegura un mínimo, a costa de renunciar a recibir el máximo de puntos posibles. En este escenario, la selección Darwiniana debería jugar a favor de las decisiones egoístas y suprimir a los cooperadores. Pero por otro lado, el sentido de conservación del ser humano y la confianza mutua puede plantear la cooperación como una buena elección permitiendo a ambos participantes ganar, siempre que los valores de c y ε sean los adecuados. En otras palabras, las decisiones individualistas se mantienen a costa de reducir las puntuaciones globales obtenidas en el sistema. Nuestro objetivo es maximizar la ganancia global.

Debido a que los usuarios son anónimos, la decisión de escoger una estrategia en vez de otra se sustenta en la confianza en el resto de participantes y su capacidad de reciprocididad. La evolución de la interacción entre los participantes se vuelve interesante cuando dos o más jugadores juegan más de una ronda, puesto que las decisiones se pueden basar en comportamientos pasados. A continuación se describen dos de las estrategias implementadas en el simulador.

Tit-for-tat es una estrategia comúnmente utilizada que consiste en que cada participante, en cada una de las rondas, escoge la misma estrategia que el resto ha escogido en último lugar con el objetivo de maximizar las ganancias individuales a costa de reducir las pérdidas. Dos aspectos fundamentales definen esta estrategia: a) no tiene memoria, ya que basa sus decisiones sin tener en cuenta qué ha ocurrido en rondas anteriores a la última; y b) para cada par de nodos, la decisión de colaborar o no se basa únicamente en la relación entre ambos nodos, sin tener en cuenta el resto de información que tiene el nodo sobre sus vecinos.

Diferentes variantes permiten a los nodos comportarse más o menos agresivamente. En nuestro caso, se ha implementado una versión neutra, en la que un nodo que recibe una petición de otro nodo de quién no posee información responderá afirmativamente con un 50% de probabilidades.

Area equilibrium es una estrategia creada especialmente para comparar el comportamiento de los nodos en el simulador. Mientras que el tit-for-tat tiene por objetivo evaluar el comportamiento de dos nodos, esta nueva propuesta intenta evaluar cada nodo frente a todos sus vecinos. Para ello, en cada una de las rondas los nodos calculan la relación entre el número de peticiones enviadas y el número de peticiones recibidas afirmativamente. La probabilidad p de aceptar una petición viene dada por la fórmula 1.

$$p(n) = 100 \left| \left(100 \sum_{t=0}^{T-1} \frac{CPU(t, \boldsymbol{v})}{CPU(t, \boldsymbol{v}) + CPU(t, n)} \right) - 50 \right| \tag{1}$$

donde p(n) representa la probabilidad de que un nodo n acepte una petición, $CPU(t, \boldsymbol{n})$ el número de slots de CPU consumidos durante un tiempo t por los nodos vecinos v_x, $CPU(t, n)$ los slots de CPU consumidos por el nodo n y T

el tiempo actual. Como resultado, si los propios recursos se han dedicado equitativamente a tareas externas y propias, la probabilidad de contestar afirmativamente es del 100%, pero si estos coeficientes no están balanceados la probabilidad se acerca al 50% linealmente. El hecho de tener el mínimo de probabilidad truncado al 50% hace ambas estrategias igualmente agresivas. Finalmente, es importante mencionar que se trata por tanto, de una estrategia con memoria.

3 Arquitectura del simulador

Para flexibilizar el uso del simulador, su arquitectura se ha separado en niveles, o capas, de responsabilidad que se encargan de aspectos concretos del sistema. Esta decisión ha permitido separar la lógica interna del simulador de la lógica que se desea probar, de forma que la redefinición de los algoritmos o protocolos a testear no influya en el funcionamiento del sistema.

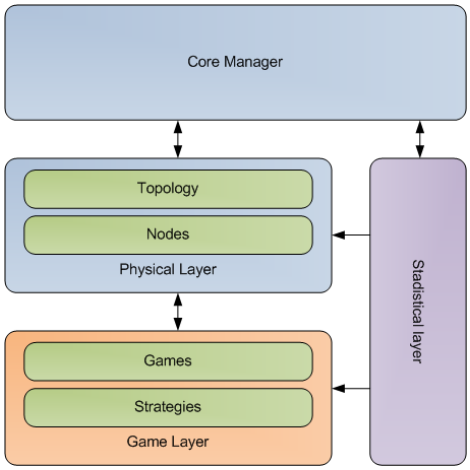

Fig. 1. Arquitectura del simulador

Como se puede observar en la Figura 1 el simulador está basado en cuatro capas, dos de bajo nivel, el core manager y la capa física; y dos de soporte que se responsabilizan de simular el juego, estrategias y recolectar los datos estadísticos necesarios. Estas dos últimas, son totalmente adaptables a las necesidades de los usuarios: la capa *Game* a través de las interfaces de clase, y la capa estratégica y de resultados a través de sus parámetros de configuración.

A continuación se muestra brevemente el funcionamiento de las dos capas de bajo nivel, mientras que en el Capítulo 4 se explorará un poco más la capa de estadísticas y resultados.

3.1 Core manager

Esta capa es la responsable de configurar y controlar la máquina de estados del simulador, así como de llamar a la lógica que contienen los métodos del resto de capas. En un primer momento, recoge los parámetros de los archivos de configuración, y carga la topología (diagrama de red y dispositivos) de la capa física e inicializa el los parámetros del módulo estadístico.

3.2 Capa de representación física

La capa de representación física únicamente contiene las descripciones del comportamiento de la red *overlay* y los diferentes tipos de nodos configurados en el sistema. También contiene los métodos y las interfaces de lectura de las diferentes topologías soportadas, actualmente BRITE [6], Pajek [7] y GEXF (*Graph Exchange XML Format*).

3.3 Algoritmo del simulador

A continuación se describe de forma genérica el funcionamiento del simulador para poder comprender mejor algunos de los resultados que se exponen más adelante. Para ello se supondrá que el simulador utiliza el juego predefinido del dilema del prisionero descrito anteriormente.

El algoritmo de simulación se divide en cuatro únicos pasos:

Peticiones: En este punto, cada uno de los nodos que no tiene tareas pendientes propias creará una nueva tarea con un 50% de probabilidades. Para poder asumirla, el nodo primero reservará sus propios recursos hasta cubrir el máximo de slots de CPU disponibles. Sin embargo, si sus recursos no son suficientes enviará una petición de CPU a cada uno de sus vecinos por la cantidad total que requiera.

Respuestas: La estrategia para responder a las peticiones es simple. Primero de todo, cada nodo que tiene una petición de reserva responderá afirmativamente a sus propias peticiones, asegurándose sus propios recursos (cabe recordar que una implementación diferente del interfaz *Game* podría cambiar este comportamiento). Después, para cada una de las peticiones de otros nodos evalúa la respuesta adecuada según la estrategia definida (nuevamente, es importante remarcar que la estrategia es definida por el usuario, y que esta puede variar o evolucionar a lo largo de la simulación). Por tanto, una petición será contestada afirmativamente si se cumplen dos condiciones: a) el nodo tiene slots disponibles y b) el resultado de la estrategia es positivo. La evaluación aleatoriamnete ordenada de las peticiones asegura un trato justo entre todos los nodos.

Evaluación: Durante esta fase cada nodo evalúa y se auto-asigna sus propias respuestas de CPU. Seguidamente, evalúa las respuestas de otros nodos y descarta el exceso de slots CPU recibido, también de forma aleatoria. No obstante, en el caso de descartar peticiones de un nodo por exceso, a éste se le considerará colaborador de cara a las próximas evaluaciones de la estrategia (al menos, en nuestra implementación del dilema del prisionero).

Guardado estadístico: Como su nombre indica, la función de esta última fase es la de recoger la información de la ronda, procesarla y guardarla para su análisis. Seguidamente, el sistema limpia las variables temporales y colas de mensajes para preparar la siguiente ronda de simulación.

4 Métricas de resultados

La capa de estadística (Ver Capítulo 3) es la responsable de calcular y guardar los resultados de todos los eventos ocurridos en el simulador. Ronda a ronda, estos datos son guardados en disco a través de archivos de resultados o bases de datos relacionales para evitar la pérdida de información en caso de algún fallo en la máquina o el simulador. La cantidad de información retenida depende del nivel de detalle deseado.

Existen, básicamente, dos ejes de datos: el espacial y el temporal. El primero referencia a cada uno de los vértices y enlaces del grafo de la topología, mientras que el segundo referencia a cada una de las rondas (o pasos) de simulación. Dentro de cada intersección, se guardan datos sobre las peticiones, el estado de los nodos o la cantidad de información que se transfiere por cada enlace. Además, se guardan datos estadísticos (mínimos, máximos, medias, correlaciones, etc) de forma global (sobre todo el conjunto de datos) o local (de cada nodo y vértice)

4.1 Interpretación de resultados

Actualmente, la interpretación de resultados se debe realizar de forma asíncrona desde programas o herramientas dedicadas externas al simulador. La interfaz *OutputCollector*, de la capa estadística proporciona una forma simple de crear objetos para la recolección de los resultados en diferentes formatos. Actualmente, el simulador proporciona clases derivadas para los siguientes formatos de salida: texto, mySQL y GEFX.

Los archivos de texto son archivos convencionales en los que, para cada fila, se ha registrado un par {clave, valores[] } separados por una tabulación. Los valores, están separados por espacios. De esta forma, pueden ser fácilmente tratados con hojas de cálculo convencionales o, más comúnmente, con herramientas de análisis basadas en patrones *MapReduce* [8]. Los otros dos formatos de salida permiten obtener representaciones de la información para entornos web o herramientas gráficas de análisis de grafos.

5 Simulaciones

Los resultados del simulador deben permitir tanto a investigadores como a diseñadores de clusters e infraestructuras de colaboración móvil conocer qué topologías, estrategias de colaboración y algoritmos de distribución de recursos dan mejores resultados.

En este capítulo se mostrarán diversos resultados, basados parcialmente en el trabajo presentado en [2]. Otros resultados pueden verse en los artículos

CSCWD'11 y MoSa'11. Así pues, el objetivo de este trabajo, más allá de presentar conclusiones y resultados útiles, es demostrar cómo nuestra herramienta es capaz de facilitar la obtención de respuestas relativas a cómo se comportan las arquitecturas de colaboración y cómo pueden mejorarse utilizando la topología.

5.1 Ratio entre dispositivos móviles y de escritorio

Esta simulación muestra el efecto de introducir ordenadores de escritorio en una red de colaboración móvil, cuando los nodos juegan una versión del dilema del prisionero utilizando la estrategia de tit-for-tat. La Figura 2 muestra el coeficiente de cooperación (porcentaje de respuestas afirmativas vs. realizadas) conseguido por los dispositivos móviles y los ordenadores personales. En estas simulaciones, los dispositivos se han situado aleatoriamente en el grafo sin ningún tipo de patrón concreto.

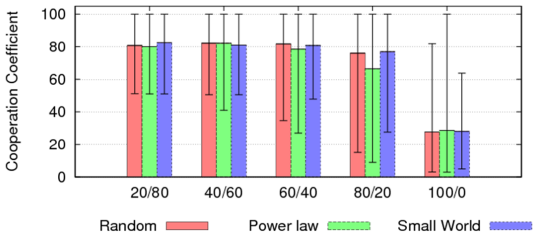

Fig. 2. Coeficiente de cooperación de los dispositivos móviles (Media, Max, Min) vs retio Móviles / PC escritorio en tres topologías diferentes

En la Figura 2 se puede observar que existen pequeñas variaciones en el coeficiente de cooperación entre redes con difertente ratio de móviles y ordenadores personales. No obstante, cuando la red está compuesta únicamente por dispositivos móviles (100/0), los valores de los coeficientes de cooperación son considerablemente más bajos. Estos resultados confirman que la introducción de los ordenadores de escritorio, independientemente de la topología de red utilizada, mejora el nivel de cooperación del sistema. Además se observa que el valor máximo de cooperación obtenido es cercano al 100% cuando el número de ordenadores personales es del 20% y aumenta de forma lineal cuando el porcentaje de ordenadores de escritorio aumenta. Incrementar este ratio mejora el valor mínimo de cooperación conseguido por el peor nodo hasta que la relación es de 60/40; a partir de la cual incrementar el número de ordenadores de escritorio en el sistema no aporta ventajas significativas.

La Figura 3 muestra la evolución del coeficiente de cooperación en una red aleatoria (Waxman) al cambiar la distribución de los nuevos recursos en los ordenadores de escritorio, manteniendo siempre la cantidad total de recursos nuevos

introducidos en el sistema. La distribución va desde un 3% de ordenadores de escritorio con unos 112-128 slots de CPU cada uno hasta un 33% de ordenadores de escritorio con 7-8 slots de CPU cada uno.

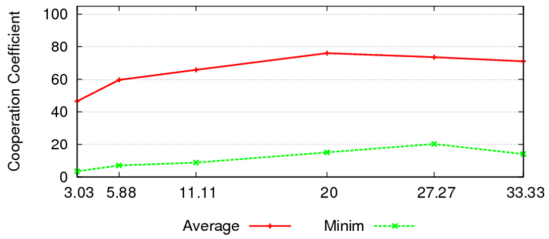

Fig. 3. Coeficiente de cooperación de los dispositivos móviles (Media, Min) vs ratio Móviles / PC escritorio en una red aleatoria

Es importante darse cuenta que la gráfica muestra dos valores de distribución de recursos óptimos. El máximo coeficiente de cooperación puede considerarse en el 20% o 27% dependiendo de la métrica utilizada (la media o el mínimo coeficiente de cooperación conseguido). En cualquier caso, es mejor tener unos pocos nodos totalmente satisfechos que una cantidad mayor de nodos parcialmente satisfechos. Por tanto, en nuestra opinión la proporción ideal se encuentra en el 27% de ordenadores de escritorio.

Los resultados descritos muestran que en un escenario donde los moviles comparten recursos es posible conseguir una optimización (máximo) de disponibilidad de recursos por nodo, si estos estan - desde un punto de vista topológico - correctamente distribuidos a través de los ordenadores de escritorio. Dicho de otra forma, podemos afirmar que los desarrolladores tienen dos variables con las que trabajar para obtener los efectos deseados: (1) el número total de nodos y (2) la cantidad de recursos disponibles. Además, el resto de estudios realizados con el simulador muestran que estos valores se pueden mejorar utilizando estrategias de posicionamiento locales basadas en: (3) distribución de los nodos en la red y (4) distribución relativa de los móviles y ordenadores de escritorio.

La Figura 4 muestra el coeficiente de reciprocidad (voluntad de cooperar) entre cada par de nodos (el eje de abscisas representa los nodos i que realizan la petición y el eje de ordenadas los nodos j que reciben la petición) en una escala del 0 hasta el 10000 dónde los valores más oscuros representa los valores más altos. Puesto que los nodos han sido ordenados en función a su coeficiente de cooperación, podemos observar fácilmente que los con valores de cooperación más altos - aquellos en la parte derecha del eje de abscisas - no corresponden con los nodos más dispuestos a colaborar - aquellos que tienen su columna más oscura. En otras palabras, los nodos más satisfechos no son necesariamente los nodos

mejor tratados por el sistema. Como consecuencia, podemos deducir que deben existir mejores criterios (en términos de justicia) para colocar cada dispositivo.

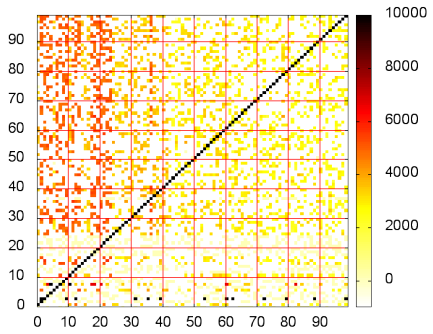

Fig. 4. Matriz de coeficiente de reciprocidad de una red Small-World con 100 nodos, 30 grados de conectividad de media vs coeficiente de cooperación.

5.2 Dependencia topológica vs estratégica

En esta simulación se muestra cómo efectivamente, la topología puede imponer un comportamiento global en los nodos de la red independietemente de la estrategia que éstos escogan cuando participan en un escenario colaborativo. En la Figura 5 se puede observar el coeficiente de cooperación conseguido por cuatro combinaciones de estrategia y topología cuando varía el número medio de grados de conectividad de los nodos.

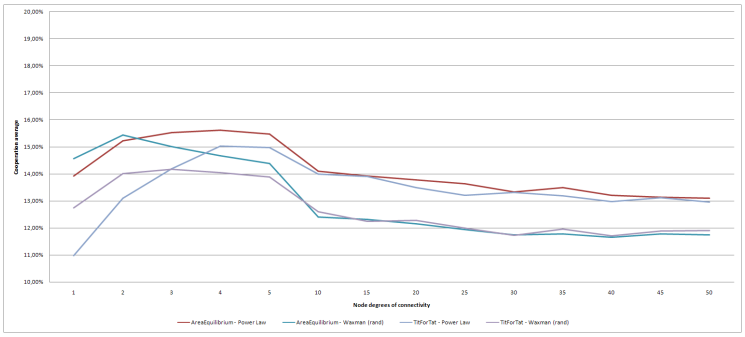

Fig. 5. Coeficiente de cooperación en diferentes redes de 1000 nodos utilizando las estrategias tit-for-tat y area equilibrium

Claramente, la simulación permite distinguir dos zonas, dependiendo de la densidad de la topología. En la zona baja (hasta 5 grados de conectividad por nodos), las simulaciones en las que los dispositivos móviles han optado por utilizar una estrategia basada en información de todos sus vecinos optienen mejores resultados que aquellos que han optado por el tit-for-tat. Ello es debido a que la voluntad de cooperación es más dificil de propagarse debido a que el radio de la red es mayor.

Sin embargo, el análisis de la parte alta (dónde las topologías tienen más grados de conectividad) muestra los resultados esperados. Independientemente de la estrategia escogida por los nodos, en la topología *Power Law* el coeficiente de cooperación medio de la red es superior, respaldando los resultados ya observados por Santos [14].

5.3 Impacto de los parámetros de la red

La información topológica de cada nodo de la topología y los resultados de las simulaciones permiten evaluar su comportamiento en función de los paámetros de la red. A modo de ejemplo, la Figura 6 muestra el coeficiente de cooperación conseguido por los móviles de una red aleatoria con 1000 nodos, 30 grados de conexión (media) por nodo cuando utilizan la estrategia de Area Equilibrium.

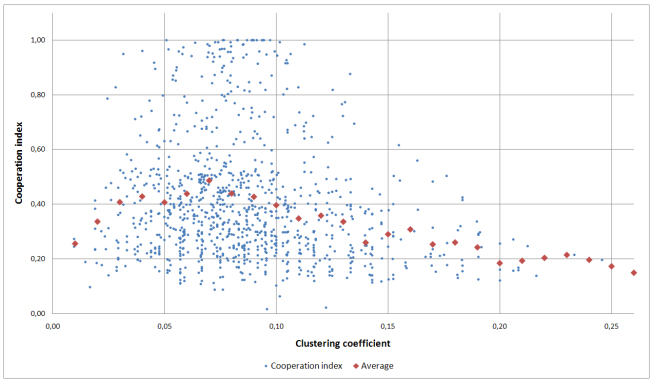

Fig. 6. Coeficiente de cooperación de los nodos vs coeficiente de clustering en una red aleatoria y homogénea de móviles con 1000 nodos

Cómo se puede observar, los resultados preeliminares del simulador parecían indicar que existía un cierto comportamiento ligado al posicionamiento de los nodos dentro de la topología. Aquellos nodos situados en posiciones con un coeficiente de clustering cercano al 0.08 obtienen mejores resultados que el resto de nodos de la topología, siendo claros candidatos a salir beneficiados. Sin embargo, aunque los resultados mostrados en la Figura 6 són correctos, posteriores intentos de reproducir este comportamiento cuando los nodos escojen otra estrategia

basada en relaciones 1:1, como el tit-for-tat, han fracasado. Esto demuestra cómo las estrategias de cooperación basadas en información de todos los vecinos (como el Area Equilibrium) son más dependientes de la topología de la red, que aquellas basadas en la relación directa entre nodos (Tit-for-tat) pese a que estas últimas dan mejores resultados globales.

En el caso del tit-for-tat ha sido necesario estudiar el comportamiento de los nodos desde un punto de vista local para poder proponer un algoritmo de posicionamiento que mejore los resultados mostrados en la Figura 2 a través de la topología. Su definición, y los resultados se pueden ver en el artículo de MoSa'11, titulado *"A Node Placement Heuristic to Encourage Resource Sharing in Mobile Computing"*.

6 Trabajo relacionado

La computación voluntaria [11] propone una idea interesante para compartir recursos hardware entre los dispositivos pertenecientes a una red overlay. Es básicamente un problema de asignación de recursos. Hay una rica literatura sobre los problemas de asignación de recursos en una amplia gama de sistemas informáticos o redes. Diferentes términos se han utilizado para este problema, pero todos ellos se refieren al proceso de elección de los recursos adecuados para alojar ciertas tareas de computación.

La asignación de recursos es un problema típico que trata de aproximar el objetivo general (de rendimiento o de mejora de los costes), la carga de trabajo y el sistema. Éste es un problema NP-Hard [16] [17], así que por lo general requiere algunas heurísticas para encontrar soluciones aproximadas en un plazo de tiempo viable. Las primeras heurísticas provienen del conocido problema *uncapacited facility location problem* [17], un problema ampliamente estudiado.

Un dominio de aplicación bien conocido de la asignación de recursos es la Web. Hay numerosos formas para colocar servidores web o servidores proxy web de una manera que se optimice el rendimiento [18] [19] [21]. Los algoritmos de asignación aplicados en todos estos casos requieren un conocimiento global acerca de la topología de la red y sobre las peticiones de los clientes. Otros enfoques se presentan en los dominios de redes *grid* [22][23] y redes *overlay* [24][25]. Estos sistemas también asumen una visión global de la red y alguna entidad de gestión centralizada. Por último, también podemos encontrar el mismo problema en nuevas áreas como entornos inteligentes (AmI). En [20] los autores proponen un algoritmo totalmente descentralizado, dinámico y adaptable para la despliegue de servicios en entornos AmI. Este algoritmo logra un patrón coordinado de asignación que reduce al mínimo los costes de comunicación sin ningún tipo de control central.

Hay algunas limitaciones conocidas de cualquier arquitectura con computación voluntaria [13]. En particular, la falta de cooperación, cuando muchos nodos se vuelven egoístas y se esfuerzan por maximizar su propia utilidad mediante el uso excesivo del sistema, sin contribuir al mismo. Un enfoque para abordar el problema de la falta de cooperación en el intercambio de recursos fue propuesta

por Feldman et al. [5]. Ellos demostraron que el mecanismo proportional-sharing logra un equilibrio razonable entre la eficiencia y el grado de equidad. Los problemas de este enfoque son su complejidad y la gran dificultad de su aplicación de manera descentralizada, con sólo información local.

Otra interesante aproximación fue propuesta por Nowak [15], que estudió las propiedades de la topología para el fomento de la cooperación en el ámbito de la teoría de juegos. En su inspirador artículo, Nowak presenta algunos mecanismos de la evolución de la cooperación y reglas muy simples que especifican cómo la selección natural puede conducir a una cooperación en un juego del Dilema del Prisionero. Estas reglas han inspirado nuestras hipótesis de trabajo. Los estudios de Cassar [12], Santos et al. [14] y Lozano et al. [4] también muestran el impacto potencial de la topología en el proceso de cooperación.

El impacto de las topologías de red overlay sobre la cooperación ya ha sido estudiado en otras áreas y con otros enfoques. Un enfoque bien conocido es el estudio de las propiedades de la topología en aplicaciones reales con un alto grado de cooperación. Iamnitchi et al. han estudiado los patrones y las propiedades de la topología de aplicaciones para compartir archivos [27], y Lozano et al. han estudiado el correo electrónico y *Pretty Good Privacy* (PGP) [4]. Las topologías utilizadas en nuestro estudio se basan en estos dos últimos trabajos.

Con todo esto parece factible de implementar estos mecanismos en escenarios móviles donde la topología de la red pueda cambiar dinámicamente y de los nodos puedan salir de la red debido a enlaces temporales o débiles. Una diferencia importante entre nuestro estudio y los estudios previos es el hecho de que nosotros modelamos la heterogeneidad y las limitaciones de los dispositivos. Nuestro estudio también tiene en cuenta las características de la red *overlay* y la colocación de los dispositivos dentro de esta red.

7 Conclusiones

El principal objetivo de este trabajo ha sido la creación de una herramienta de análisis de comportamientos sociales en entornos de computación distribuida dónde participan dispositivos heterogéneos. Nuestra atención se ha centrado en resolver el problema de compartición de recursos físicos (ciclos de CPU) desde un punto de vista topológico.

Nuestro estudio se ha soportado a través de varias simulaciones, en las cuales nuestra herramienta ha permitido analizar el impacto de la distribución de los dispositivos móviles en varias redes *overlay* utilizando modelos simples de nodos móviles y ordenadores personales. Como resultado, se han obtenido una serie de observaciones empíricas qeu permiten establecer reglas simples para los diseñadores de *clusters* móviles a la hora de incentivar a los usuasrios a comaprtir sus recursos computacionales:

Ratio entre móviles y ordenadores de escritorio. En base a nuestra simulaciones, es posible concluir que la cantidad de recursos que deben aportar los dos tipos de dispositivos deben ser del mismo orden de magnitud. El coste económico

de incrmenentar este ratio en favor de los ordenadaores de escritorio no compensa.

Topología vs estrategia. Parte de nuestros resultados muestran cómo el comportamiento de los nodos puede ser controlado o guiado escogiendo adecuadamente topologías concretas. Por desgracia, las deciciones topológicas a nivel global han demostrado ser insuficientes para sacar provecho de este fenómeno; siendo necesaria una estrategia de mejora individual.

El trabajo futuro se centrará en explorar métodos inteligentes que permitan al simulador establecer reglas de comportamiento, gracias a las cuales los nodos pueden rehubicarse en tiempo real para mejorar su posición en la red. Para ello, será necesario redefinir el protocolo de negociación y el algoritmo de posicionamiento propuesto para los móviles, de forma que solo requiera información local.

Agradecimientos. Este trabajo ha sido parcialmente financiado por el MICINN, proyecto DELFIN, TIN2010-20140-C03-01.

References

1. Pinelle, D.; Gutwin, C.: Loose Coupling and Healthcare Organizations: Deployment Strategies for Groupware. Computer Supported Cooperative Work Journal 15, 5-6, 2006, 537-572.
2. Vega, D.: Design and implementation of a simulator to explore cooperation in distributed environments. Master thesis, Universitat Politècnica de Catalunya, Spain, 2010
3. Lai, K.; Rasmusson, L.; Adar, Sorkin, S.; Zhang L.; Huberman B. A.: Tycoon: an Implemention of a Distributed Market-Based Resource Allocation System REVISAR
4. Lozano, S.; Arenas, A.; Sánchez, A.: Mesoscopic Structure Conditions the Emergence of Cooperation on Social Networks. PLoS ONE, Public Library of Science, 2008, 3, e1892
5. Feldman, M.; Lai, K.; Zhang, L.: The Proportional-Share Allocation Market for Computational Resources. IEEE Transactions on Parallel and Distributed Systems, IEEE Computer Society, 2009, 20, 1075-1088
6. Medina, A.; Lakhina, A.; Matta, I.; Byers, J.: BRITE: an approach to universal topology generation. Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOT), 2001, 346 -353
7. Batagelj, V.; Mrvar, A.: Analysis and Visualization of Large Networks. Graph Drawing Software. Springer, Berlin 2003. p. 77-103 / Amazon.
8. Dean, J.; Ghemawat S.: MapReduce: Simplified Data Processing on Large Clusters. OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004
9. FIPA Contract Net Interaction Protocol Specification. FIPA (2002, March 12) In: Foundation for Intelligent Physical Agents [On line] Standard SC00029H [Retrieved: June 14, 2010]. Available at: ¡http://www.fipa.org/specs/fipa00029/SC00029H.pdf¿

10. FIPA Communicative Act Library Specification. FIPA (2010, March 12) In: Foundation for Intelligent Physical Agents [On line] Standard SC00037J [Retrieved: June 6, 2010]. Available at: ¡http://www.fipa.org/specs/fipa00037/SC00037J.pdf¿
11. Sarmenta, L. F. G.; Hirano, S.: Bayanihan: building and studying web-based volunteer computing systems using Java. Future Generation Computer Systems, 1999, 15, 675 – 686
12. Cassar, A.: Coordination and cooperation in local, random and small world networks: experimental evidence. Games and Economic Behavior, 2007, 58, 209 – 230
13. Cusack, C.; Martens, C.; Mutreja, P.: Volunteer Computing Using Casual Games. Future of Game Design and Technology (FuturePlay), 2006
14. Santos, F. C.; Rodrigues, J. F.; Pacheco, J. M.: Graph topology plays a determinant role in the evolution of cooperation. Proceedings of the Royal Society B: Biological Sciences, 2006, 273, 51-55
15. Nowak, M. A.: Five Rules for the Evolution of Cooperation. Science, 2006, 314, 1560-1563
16. Chevaleyre, Y.; Endriss, U.; Lang, J.; Maudet, N. van Leeuwen, J.; Italiano, G.: A Short Introduction to Computational Social Choice. Theory and Practice of Computer Science (SOFSEM), 2007, 4362, 51-69
17. Cornuejols, G. P., Nemhauser, G. L., Wolsey, L. A. X.: The uncapacitated facility location problem. Discrete Location Theory, Wiley, 1990, 119–171.
18. Coppens J., Wauters T., De Turck F., Dhoedt B., Demeester P.: Evaluation of replica placement and retrieval algorithms in self-organizing CDNs, Proc of IFIP/IEEE International Workshop on Self-Managed Systems & Services SelfMan, 2005.
19. Karlsson M., Mahalingam M.: Do We Need Replica Placement Algorithms in Content Delivery Networks? Proc Web Content Caching and Distribution Workshop, 2002.
20. Herrmann, K.: Self-organized service placement in ambient intelligence environments. ACM Trans. Auton. Adapt. Syst., ACM, 2010, 5, 6:1-6:39
21. Tang, X.; Chi, H.; Chanson, S. T.: Optimal Replica Placement under TTL-Based Consistency IEEE Transactions on Parallel and Distributed Systems, IEEE Computer Society, 2007, 18, 351-363
22. Lee, B-D.; Weissman, J.: Dynamic replica management in the service grid Proc High Performance Distributed Computing (HPDC)., 2001, 433 -434
23. Graupner, S.; Andrzejak, A.; Kotov, V.; Trinks, H. Brueckner, S. A.: Adaptive Service Placement Algorithms for Autonomous Service Networks Engineering Self-Organising Systems, 2005, 3464, 331-337
24. Liu, K. Y.; Lui, J. C.; Zhang, Z.-L. Mitrou, N.: Distributed Algorithm for Service Replication in Service Overlay Network Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications (NETWORKING), 2004, 3042, 1156-1167
25. Choi, S.; Shavitt, Y.: Placing servers for session-oriented services Department of Computer Science, Washington University. Tech. rep. WUCS-2001-41, 2001.
26. Legout, Arnaud Clustering and sharing incentives in bittorrent systems In SIGMETRICS'07, 301–312, 2006
27. Iamnitchi, A.; Ripeanu, M.; Foster, I.: Small-world file-sharing communities. IEEE Computer and Communications Societies (INFOCOM), 2004, 2, 952 - 963 vol.2