

Patrick Griffin

STS.035: Historia de la computación

Artículo de respuesta a las lecturas de la semana 11

Perspectivas de futuro: Computación en rejilla frente a computación fiable

Recientemente, IBM y otras compañías, han promovido una nueva ola de computación, en la que multitud de ordenadores se combinan para formar una imagen de sistema única, que permite que los usuarios del sistema ejecuten programas distribuidos en muchas máquinas o que accedan de forma uniforme a los archivos distribuidos por todo el mundo. Esta perspectiva de “computación en rejilla” prevé que los ciclos de procesador estarán alquilados por toda la red, así que la potencia computacional se convertirá en un servicio, como pueden serlo el agua o el servicio eléctrico. Entretanto, muchas sociedades anónimas fomentan también la “computación fiable”, que permite a los ordenadores remotos verificar la ejecución de algún programa en una máquina local. Aunque ambos esquemas tienen como núcleo a Internet, los dos sistemas tienen propósitos muy distintos para la interacción final entre el usuario y sus ordenadores.

La computación en rejilla facilita mecanismos para la unificación de muchos ordenadores en un sistema único y más manejable. Quizás el proyecto de computación en rejilla más importante hasta la fecha es el lanzamiento del kit de herramientas Globus (Waldrop pág. 33), que proporciona muchas de las utilidades y protocolos necesarios para lograr crear tan enorme sistema. Lanzado como un proyecto de fuente abierta, Globus se ha convertido en la base de la mayoría de los sistemas de computación en rejilla. Siguen existiendo dificultades, como por ejemplo, la asignación de programas a las máquinas que mejor satisfacen sus necesidades, lo cual es un desafío fundamental para los simuladores del plegamiento de proteínas (Johnson 3). Sin embargo, la computación en rejilla tiene como objetivo ofrecer una imagen computacional uniforme a todos los usuarios, con programas y recursos accesibles.

La computación fiable proporciona mecanismos para verificar la ejecución de un programa en una única máquina. Autentifica el código del programa en toda la red, asegurando que el programa que se está ejecutando en la máquina local está autorizado por alguna autoridad remota. Existen muchas aplicaciones de computación fiable, entre las que se incluyen la protección de virus y la gestión de los derechos digitales. De hecho, la capacidad de confiar en que un ordenador concreto ejecute un cierto programa adecuadamente, es decisivo para la computación en rejilla. No obstante, las compañías que se dedican a la computación fiable suelen contemplarla como un mecanismo para adaptar el software a las máquinas individuales, de modo que un programa concreto o algunos datos puedan utilizarse únicamente en una máquina. Utilizada de este forma, la computación fiable permite que las compañías de software y de medios de comunicación reduzcan la piratería, mediante la certificación individual de cada copia del programa o medio. Por tanto, podemos observar la distinción entre los objetivos de estos dos tipos de tecnología –los promotores de la computación en rejilla han dado publicidad a un sistema abierto en el que cada nodo tiene la misma interfaz visible al usuario que cualquier otro, mientras que los promotores de la computación fiable esperan controlar una red diferenciando cada ordenador de usuario final.

Aunque la tecnología implicada en estas dos perspectivas sea fundamentalmente la misma –computación procesada por la distribución de la información a través de Internet– los objetivos de sus creadores conllevan dos puntos de vista sociales distintos. La computación en rejilla pretende crear un mundo de información que se extiende sobre recursos globalmente accesibles. La seguridad debe garantizarse, por supuesto, pero su sueño es permitir, por ejemplo, que los ecologistas ejecuten las mismas simulaciones que los legisladores políticos, al decidirse por un proyecto de

desarrollo. (Waldrop p. 7). Por lo tanto, la computación en rejilla contempla un mundo en el que el hombre interactúa con un ordenador “en rejilla” global, en vez de hacerlo con máquinas aisladas e individuales. La computación fiable permitiría además, que la información se expandiese por una red con garantías de seguridad, pero sus promotores están centrados en utilizar la tecnología para evitar que la ejecución de un programa se lleve a cabo en una sola máquina. Dichas características serían útiles, por ejemplo, para garantizar que los individuos adquieran una copia nueva de un programa para cada ordenador en el que éste se ejecute, o para evitar la batalla de la piratería de archivos musicales digitales especialmente codificados. No obstante, esta visión conduce a una dinámica de usuario-ordenador distinta a la de la computación en rejilla: en vez de permitir un acceso uniforme para todos, los programas sólo pueden codificarse para ejecutarse en máquinas concretas, mientras se sigue permitiendo que éstos se transmitan a través de la red. Por tanto, mientras la computación en rejilla trata de compartir los recursos computacionales en gran escala, muchos promotores de la computación fiable esperan localizar los recursos, de manera que puedan utilizarlos “de forma segura”, como alternativa a la red.

Las consecuencias sociales de cada uno de estos modelos aún no se han vislumbrado. Una perspectiva utópica de la computación en rejilla presupone que los datos y recursos computacionales se compartirán para consumo público. Una de las visiones más negativas de la computación fiable localiza a los ordenadores junto con su software, de modo que sólo una máquina en concreto pueda acceder a los datos, generando lógicas dificultades para la conservación histórica y la creación de un dominio público de información. En realidad, se pondrá en funcionamiento una mezcla de estas dos perspectivas tecnológicas –la computación en rejilla no madurará a menos que los usuarios confíen en ella, y la computación fiable no tiene por qué utilizarse para restringir información, como en su aplicación actual planificada para la licencia de software y la gestión de derechos digitales.

Bibliografía

Johnson, George. “Supercomputing ‘@Home’ Is Paying Off”. The New York Times (23 de abril de 2003): F1

Waldrop, M. Mitchell, “Grid Computing”. Technology Review (mayo de 2002):31-37.

Trusted Computing Platform Alliance. <http://trustedcomputing.org> (abril de 2004).