

Transferencia Masiva e Intensiva de Datos en Computación de Alto Rendimiento

Intensive and Masive Data Transfer in High Performance Computing

Carlos Jaime BARRIOS-HERNÁNDEZ

Laboratoire d' Informatique de Grenoble (LIG), Equipo MESCAL

Laboratoire d' Informatique, Signaux et Systèmes de Sophia-Antipolis (I3S), Equipo RAINBOW
Francia

barrios@imag.fr

Abstract

Today, in spite of the technological advances, the data transfer in high performance architectures is a critical factor. Different behaviors of applications in execution can be observed in similar platforms. Also when the applications are deployed in distributed systems like a Grid infrastructure. This paper¹ presents a description of this problem and it propose many points of view of traitment and different strategies of test.

Resumen

La transferencia de datos en arquitecturas de computación de alto desempeño es un factor crítico a pesar de los avances tecnológicos. Diferentes comportamientos de aplicaciones en ejecución son observados en plataformas similares y más aún, al desplegarla sobre arquitecturas de gran escala, como lo es una infraestructura GRID. Este artículo² presenta una descripción de esta problemática interesante y propone puntos de tratamiento y experimentación.

1. Introducción

Es bien conocido que uno de los factores críticos al momento de ejecutar una aplicación paralela, es el tiempo de comunicación. Diferentes autores han propuesto modelos de análisis de transferencia de datos en aplicaciones paralelas, de acuerdo a los puntos de vista:

¹This paper is a production of the ECOS C07M02 action between the INPG (France) and the UniAndes and UIS Universities (Colombia).

²Este artículo hace parte de la producción de la acción ECOS C07M02 comprendida entre el Instituto Nacional Politécnico de Grenoble-INPG (Francia), la Universidad de Los Andes de Bogotá-UniAndes y la Universidad Industrial de Santander en Bucaramanga-UIS (Colombia).

- *Físicos/Tecnológicos*: Tipo de Red, Topología de Red, Tecnología de Red
- *Lógicos/Tecnológicos*: Algoritmos de Comunicación, Protocolos de Transferencia
- *Tipo de Datos/Tipo de Aplicación*: Transferencia Masiva de Datos (Mensajes de gran tamaño), Transferencia Intensiva de Datos (Grantidad de Mensajes Transferidos)

Normalmente, el primer punto de vista, se trata más entre los especialistas en redes y telecomunicaciones que entre la comunidad de computación de alto desempeño. Sin embargo, debido a la computación GRID, la comunidad de la Computación de Alto Rendimiento ha venido abordando el tema. Los otros dos puntos, se relacionan fuertemente con el diseño de aplicaciones y el mismo despliegue y coordinación de recursos, tanto para aplicaciones paralelas que se ejecutan en clusters, como para aplicaciones que se esperan sean distribuidas en infraestructuras de gran escala, es decir, una infraestructura Grid

De la misma manera hay que observar que existen otras características que pueden incluirse simultáneamente en los tres puntos de vista tratados, como es el tipo de arquitectura (Cluster, Grid), paradigma (datos, procesos), heterogeneidad de recursos, dinamicidad, entre otros.

En este artículo, se presenta una descripción de la transferencia de datos teniendo en cuenta los tres puntos de vista propuestos. Una presentación conceptual de la problemática ha sido presentada en la sección uno que comprende esta introducción, posteriormente se presentarán algunos modelos matemáticos propuestos para el análisis de transferencia de datos tanto en clusters como en grids en la sección dos. En la sección tres, se presenta propuestas de análisis experimentales y los problemas abiertos presentes y en la sección cuatro, las perspectivas de investigación posibles a abordar y finalmente las conclusiones.

2. Algunos Modelos de Transferencia

La transferencia de datos, en un modelo de comunicación simple y ampliamente conocido, es el proceso en el cual se transfiere un paquete desde un emisor hacia un receptor utilizando un canal de comunicación, a ese paquete se le denomina mensaje. El proceso de comunicación implica un tiempo y normalmente, el paquete contiene algo que tiene un tamaño o peso. Con este simple análisis, se propuso la simple expresión de tiempo de comunicación en la transferencia de un mensaje m , bien conocida que es:

$$T_C(m) = T_E + T_T + T_R \quad (1)$$

donde $T_C(m)$ es el tiempo de comunicación en cuestión, T_E es el tiempo que toma el mensaje de salir del emisor, T_T es el tiempo que transcurre cuando el mensaje "viaja" por el canal de comunicación y T_R el tiempo de recepción.

Hasta este punto, todo parece muy sencillo y resuelto. Pero, si establecemos comunicaciones que dependen unas de otras, que exigen una sincronización, si igualmente, no es una única comunicación sino múltiples comunicaciones que se hacen de un emisor hacia varios emisores o viceversa, o entre múltiples emisores y receptores... de manera consecutiva, dinámica y entre emisores y receptores que tienen características diferentes..., con mensajes que varían de tamaño y utilizan el canal de diferentes maneras..., que pasaría con el modelo simple?.

A partir de estas situaciones se han propuesto diferentes modelos de transferencia. Sólo se tratarán algunos que se encuentran dentro del contexto de la computación de alto desempeño, teniendo en cuenta características como la simplicidad y facilidad de validación a través de la experimentación, de acuerdo a los tres puntos de vista presentados en la introducción.

Antes de presentar algunos modelos para las situaciones anteriormente descritas, es importante tratar el uso, la capacidad y la disponibilidad del medio de comunicación en una transferencia. Si realizamos una abstracción de un flujo de transferencia, podemos representarla como en la figura 1 en la cual el *flujo de datos* pasa a través de un *tubo*. La cantidad de ese flujo que representa el uso, puede observarse en términos de latencia y la capacidad del tubo en términos de ancho de banda. Obvio, esa relación entre uso y capacidad expresa la disponibilidad.

Más formalmente se conoce la *latencia*, como el tiempo que tarda un mensaje de ir desde un punto a otro, por un canal definido. En cuanto al *ancho de banda* es la cantidad de datos transferidos por unidad de tiempo. La capacidad, definida en este contexto es la cantidad de datos máxima que puede ser transferida por el canal en unidad de tiempo. La utilización, también podría expresarse como el tráfico global que existe en el canal, así que de la misma manera esta fuertemente ligada al ancho de banda.

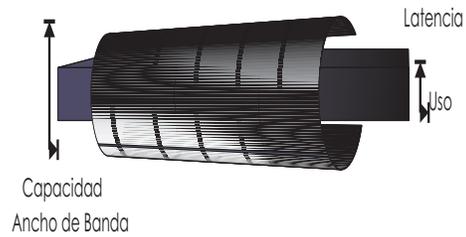


Figura 1. Abstracción del Ancho de Banda y Latencia

Los modelos físico-matemáticos y/o lógico-matemáticos que se presentan normalmente³, se expresan en términos de latencia y ancho de banda, pues precisamente buscan describir la capacidad, el uso, la disponibilidad de los recursos implicados en la transferencia y predecir las posibilidades de comunicación. La expresión más conocida que relaciona estas dos características es:

$$T = \alpha + \frac{x}{\beta} \quad (2)$$

donde T es el tiempo que se necesita para enviar s bytes de datos por un enlace en una red cualquiera, α es la latencia y β el ancho de banda [5].

En este punto es necesario contextualizar un poco la transferencia en computación de alto rendimiento. La ejecución de programas paralelos implica una transferencia igualmente de datos entre los nodos (o procesadores) de una infraestructura paralela. Esa transferencia, sea realizada por ejemplo en un cluster o en un grid, tiene características propias de acuerdo al tipo de red y gestión de recursos y por ejemplo, puede estudiarse como una transferencia concurrente. Teniendo en cuenta esto, pensamos que se pueden identificar características de red (como el ancho de banda y la latencia) por enlace entre un par de nodos y otras características de red globales o compartidas, como el ancho de banda compartido, que comunmente se representa con la siguiente expresión:

$$B_c = \beta * (N + n) \quad (3)$$

La expresión 3 es una expresión altamente experimental, donde B_c es el ancho de banda compartido, β es el ancho de banda por conexión, N la cantidad de conexiones abiertas utilizadas por la aplicación que estamos usando para realizar la medida y n son las conexiones usadas por otras aplicaciones [10] [5].

³Existen otras posibilidades, como modelos heurísticos que son altamente complejos pero igual muy eficientes para describir ciertas características y comportamientos específicos, tanto a nivel de aplicaciones en ejecución como a nivel de la infraestructura de red, que no serán tratados en este artículo.

2.1. PRAM

PRAM⁴ es un modelo de máquina abstracta para diseñar algoritmos dirigidos a computadores paralelos. PRAM es una extensión del modelo RAM que es dirigido hacia computadores secuenciales. Una descripción completa del modelo puede estudiarse en [1], pero en este artículo se presentarán algunos aspectos teóricos importante, de manera conceptual.

PRAM se caracteriza porque que elimina el enfoque hacia puntos como la sincronización y la comunicación, para explotar explícitamente la concurrencia. En términos de la Taxonomía de Flynn, PRAM se puede clasificar hacia máquinas MIMD⁵.

La operación de un PRAM sincrónico se expresa como accesos simultáneos por múltiples procesadores hacia una misma memoria compartida. Existen muchas variantes del modelo PRAM, de acuerdo a implementaciones explícitas, pero normalmente estas dependen de los accesos simultáneos que son permitidos o prohibidos. Los accesos, como en muchos otros modelos son definidos como lecturas o escrituras y para PRAM existen las siguientes posibilidades:

- *EREW* (Exclusive Read Exclusive Write): Lectura Exclusiva, Escritura Exclusiva, en la cual cada celula de memoria puede ser leída unicamente por un procesador al mismo tiempo.
- *CREW* (Concurrent Read Exclusive Write): Lectura Concurrente, Escritura Exclusiva, en la cual múltiples procesadores pueden leer una celula de memoria pero sólo uno puede escribir al tiempo.
- *ERCW* Exclusive Read Concurrent Write): Lectura Exclusiva, Escritura Concurrente, la cual es la contraria a la anterior y nunca es considerada pues su implementación no tiene sentido.
- *CRCW* (Concurrent Read Concurrent Write): Lecturas y Escrituras Concurrentes, donde los múltiples procesadores pueden escribir y leer. De esta variante existen otras posibilidades que son:
 - *CCRCW* (Common CRCW): CRCW Común, en el cual, si los procesadores escriben el mismo valor se toma como permitido o correcto y en caso contrario como una operación ilegal.
 - *ACRCW* (Arbitrary CRCW): Si al menos una tentativa es acertada, las otras se retiran.
 - *PCRCW* (Priority CRCW): Se da una prioridad a una de las escrituras, de acuerdo a un rango pre-determinado.

⁴Parallel Random Access Machine.

⁵Multiples Instrucciones Multiples Datos.

Aunque existen diferentes códigos y pseudocódigos de PRAM y algunas implementaciones experimentales, estas no han sido realmente eficientes, pues aunque PRAM permite entender la concurrencia, dividiendo el problema original en subproblemas similares y resolverlos en paralelo, la eliminación de la sincronización y el análisis de comunicación no permiten que PRAM sea realmente eficiente y normalmente se apoya en otros modelos para resolver los problemas asociados con la transferencia.

2.2. BSP

Al igual que con PRAM, solo se presentará una descripción conceptual. El Modelo de Paralelismo Sincrónico Generalizado o BSP⁶[4] es un modelo de programación paralela que representa las estructuras de programa desde un bajo nivel en favor de una entidad conocida por ese modelo como "super-pasos"(supersteps). Un superpaso es un conjunto de cálculos locales independientes, dirigidos por una comunicación global y una barrera de sincronización.

Los programas estructurados de esta manera, permiten establecer costos que pueden ser determinados desde los parámetros arquitecturales simples, identificando desde la permeabilidad de la red de comunicación hasta el tráfico uniforme-randómico y el tiempo para sincronizar. Algo interesante de este modelo es que contrario a otros modelos, las barreras de comunicación y las permutaciones de enrutamiento son observadas como situaciones no costosas. Como resultado, la estructura propuesta por BSP no reduce el desempeño, lo cual es beneficioso para una análisis de una aplicación, tanto desde el punto de vista del uso, como de la construcción de la misma [4].

2.3. Modelos Parametrizados: LogP y sus derivados

Como es bien conocido, una manera útil de representar un fenómeno modelado, es poder definir las variables que lo describen a través de parámetros. Esos parámetros, claramente identificados, permiten una fácil captura de sus valores por medios experimentales y los cálculos asociados son relativamente simples.

LogP [6] es un modelo parametrizado que permite el diseño y análisis de algoritmos asociados a aplicaciones paralelas, basado en el desempeño de la comunicación y las características de transferencia. La principal ventaja de este modelo es su simplicidad y esto ha permitido la derivación de otros modelos, teniendo en cuenta nuevas características de transferencia, algunas de estas derivaciones serán descritas en este artículo.

LogP, esta basado en cuatro parámetros que son una abstracción del ancho de banda presente en la comunicación,

⁶Bulk Synchronous Parallelism

el retardo en la transferencia y la eficiencia del acomplamiento entre la comunicación y la ejecución de la aplicación. Estos cuatro parámetros entonces son, la Latencia L , el costo de envío y recepción de un mensaje O , el espacio que existe entre la transferencia de dos mensajes consecutivos gap y la cantidad de procesadores o nodos implicados P . Todos en función del tamaño del mensaje m . Este modelo LogP es propuesto para mensajes de tamaño pequeño.

Desde LogP, es propuesto LogGP [2] que incorpora el modelo LogP en la transferencia de mensajes de gran tamaño. El principio de incorporación es igualmente simple, pues agrega un quinto parámetro denominado G que esta asociado al espacio entre pedazos de un paquete de gran tamaño que es dividido al momento de hacer la transferencia. Es decir, es el espacio por byte en la transferencia de mensajes de gran tamaño.

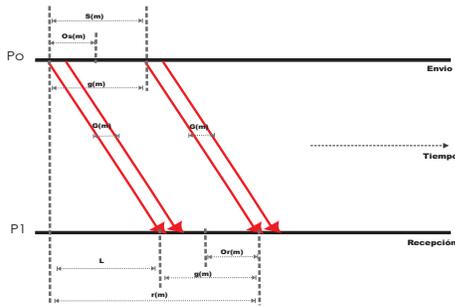


Figura 2. *Parámetros de LogGP*

En la figura 2 se representan los parámetros en la transferencia de un mensaje de gran tamaño m entre un par de procesadores P_0 y P_1 . Al enviar dos mensajes consecutivos en una experiencia de transferencia, se puede capturar y medir el tiempo que toma en ser enviado el mensaje $S(m)$ y más específicamente el costo de tiempo que representa la salida del mensaje completamente desde el procesador emisor $o_s(m)$ e igualmente ese mismo costo en terminos de recepción $o_r(m)$ en el tiempo de recepción del mensaje $r(m)$. El espacio de tiempo que existe entre dos mensajes de la misma talla enviados en la experiencia es representado por el gap $g(m)$ y el espacio entre los paquetes del mensaje de gran tamaño es $G(m)$. A partir de estos valores capturados, puede calcularse (o igual medirse) la latencia $L(m)$ y el ancho de banda $\beta(m)$ que no se representa en la figura. La captura de estos parámetros y una amplia discusión sobre su aplicabilidad es presentada en [14], y un ejemplo de utilización para el análisis de desempeño en clusters y grids utilizando este modelo es presentada en [3].

En la tabla 1 se presentan los parámetros de LogGP expresados en términos de LogP [14].

Otra de las derivaciones interesantes de $LogP$ es la denominada $LoGPC$ propuesta en [17]. Uno de los proble-

LogP/LogGP	LogP parametrizado
L	$= L + g(m) - o_s(m) - o_r(m)$
o	$= (o_s(m) + o_r(m))/2$
g	$= g(m)$
G	$= g(m)/m$, para un mensaje de gran talla
P	$= P$

Cuadro 1. *LogGP expresado en términos de LogP*

mas que tienen las aplicaciones paralelas es la contención en la comunicación. El modelo, es altamente aplicable a comunicaciones distribuídas en infraestructuras como grid, y al agregar el parámetro de contención C_n tiene en cuenta situaciones como transferencias sincrónicas y asincrónicas y saturaciones generadas en la red al momento de la comunicación. Un modelo asociado es el modelo $LoPC$ que se basa en la contención de la aplicación en sí, fuertemente relacionada con la transferencia de mensajes de gran tamaño. Este modelo se presenta en [9].

3. Análisis Experimentales

El modelamiento, en ciencias de la computación se propone para analizar una aplicación o sistema existente o para especificar una aplicación o sistema que va a ser diseñado o construído. En el contexto de este artículo, es usado para la evaluación de desempeño de la transferencia de datos másivos e intensivos en una infraestructura determinada. Esa *evaluación* puede realizarse, ya sea por medio de la simulación o por medios experimentales.

En ambos casos existen ventajas y desventajas. Proyectos como SimGrid[22] y GridSim[12] tratan la problemática desde el punto de vista de la simulación, mientras que otros buscan definir metodologías de experimentación para la validación de modelos asociados a aplicaciones paralelas, tanto en clusters como en grids. En este artículo solo nos referiremos a la parte experimental, sin definir algún tipo de metodología *excepcional* pero si nombrando los diferentes aspectos que hay que tener en cuenta para analizar el desempeño de las aplicaciones en infraestructuras de computación de alto desempeño⁷.

3.1. Aspectos Tecnológicos

En computación de alto rendimiento, tenemos dos escenarios de ejecución de aplicaciones desde un punto de vista

⁷E incluso puede extenderse hacia la evaluación de la arquitectura misma.

de escala de la infraestructura, el primero es si estamos ejecutando la aplicación en una máquina paralela única, como un cluster y el segundo si lo que estamos haciendo es ejecutando la aplicación en una infraestructura distribuida, como la grid. En términos más específicos, en el caso de la computadora paralela, se tiene en cuenta factores tecnológicos como el tipo de máquina paralela, clasificada por ejemplo dentro de la clasificación arquitectural de Flynn, pero en el caso de la grid, es necesario hablar en términos de cooperación y los factores están fuertemente asociados, como se mencionó anteriormente a las características de la red.

En ambos casos es necesario observar y tener en cuenta las características individuales de los dispositivos de red y de los nodos que la componen. Otros aspectos como la gestión que hace cada máquina de esos dispositivos, la gestión del sistema de archivos desde un punto de vista más lógico, las características del tipo de red y los protocolos usados en la transferencia, deberán ser tenidos en cuenta igualmente. Por ejemplo, una red basada en *Myrinet* [18] tiene un comportamiento muy diferente a una basada en *Infiniband* [13], teniendo en cuenta únicamente una de sus características, por ejemplo en el caso de *Myrinet*, el procesamiento de las comunicaciones de red se hace a través de chips integrados en las tarjetas de red de *Myrinet* (Lanai chips), descargando a la CPU de parte del procesamiento de las comunicaciones; mientras que en el caso de *Infiniband* usa un bus serie bidireccional de tal manera que evita los problemas típicos asociados a buses paralelos.

Normalmente, los valores asociados a los dispositivos son dados por los fabricantes y son conocidos, por lo que se toman directamente al momento de experimentar sobre una aplicación determinada. Quizás, uno de los líos es el indeterminismo al momento de ejecutar una aplicación paralela sobre un protocolo de transferencia determinado. Por ejemplo, si tenemos en cuenta TCP, nos veremos enfrentados a diferentes situaciones aún en estudio para computación de alto desempeño. Una discusión al respecto es presentada en [7].

3.2. Aspectos Arquitecturales

Los aspectos arquitecturales igualmente están fuertemente ligados a los aspectos tecnológicos. Pero en este artículo nos referiremos únicamente a lo asociado a la topología de red y a la organización de los procesadores. Si se parte que una transferencia cluster es más *directa* por denominarla de alguna manera, que una transferencia grid, inmediatamente tenemos un factor de diferenciación que sugiere un análisis diferente. En el caso de un cluster, por ejemplo, es más la organización de los procesadores en la infraestructura que afecta el desempeño de la comunicación que por ejemplo, el tipo de red utilizada, pues estamos hablando de una infraestructura homogénea y en fuertemente acopla-

da. En el caso de una infraestructura grid, los componentes están distribuidos geográficamente, son en la mayoría de casos heterogéneos y las topologías de red pueden variar de una plataforma local a otra.

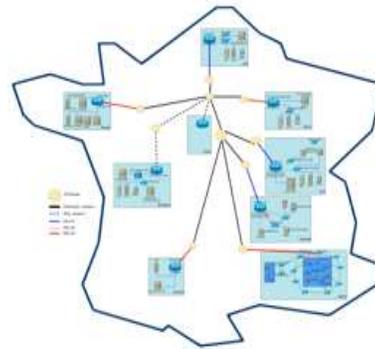


Figura 3. Grid 5000 Infraestructura

La figura 3 muestra la infraestructura del proyecto francés Grid 5000[11] desde un punto de vista de arquitectura de red. En la gráfica claramente se observa que tanto arquitecturalmente como tecnológicamente la infraestructura varía.

3.3. Otros Aspectos

Entre los otros aspectos a tener en cuenta están aquellos que se tienen en cuenta desde el punto de vista lógico y el punto de vista humano.

Desde el punto de vista lógico hay que observar el tipo de comunicación usada en los algoritmos de comunicación implantadas ya sea por la librería de paso de mensajes en el caso del uso de ese paradigma o en la aplicación misma. Por ejemplo, existe una gran diferencia entre el uso de comunicaciones bloqueantes y no bloqueantes en una aplicación determinada. La gestión de recursos y el tipo de mensajes generarán un tipo de comunicación determinada. Por ejemplo, si es intensiva, es posible que la múltiple consulta y transferencia de datos, así sea de mensajes de tamaño pequeño, genere un uso de recursos mayor que si se transfiriera un único mensaje de talla superior. O por ejemplo, la distribución de mensajes de gran tamaño entre los recursos durante la ejecución de una aplicación, puede generar saturación en la red.

Desde el punto de vista humano, que es aún un tema de discusión muy dinámico, el rendimiento de una infraestructura determinada se ve afectado por la misma cantidad y tipos de utilización de una plataforma determinada o de una infraestructura distribuida. La cancelación de procesos pueden bloquear nodos y afectar redes. La reservación y no uso de recursos así mismo limita las posibilidades de despliegue y es un factor que está asociado más a la utilización

de los recursos por los humanos que a las propias estrategias de gestión implementadas en administradores de recursos.

Un aspecto que no ha sido clasificado en una sección especial en este artículo, pero que es importante sería el aspecto metodológico al momento de realizar la experimentación. Tradicionalmente, se observa estadísticamente tras muchas pruebas el comportamiento tanto de la infraestructura como de la aplicación en situaciones determinadas. Esas situaciones van desde las situaciones mínimas hasta las situaciones extremas, para identificar los límites del desempeño. Una de las condiciones límites, propuestas incluso desde el modelamiento matemático mismo, son las condiciones de saturación de los recursos, ya sea por el tamaño del mensaje, por la cantidad de mensajes emitidos o por el número de solicitudes asociadas a usuarios que bloquearían el sistema, lo cual en ciertos casos no es aconsejable, pues daría información falsa. Muchos proyectos están dirigidos hacia la proposición de metodologías de experimentación y evaluación de desempeño de aplicaciones e infraestructuras, de acuerdo a diferentes casos de utilización.

4. Perspectivas

Sin duda el análisis de desempeño de aplicaciones paralelas y distribuidas desde el punto de vista de comunicación plantea aún situaciones a resolver, independiente de que sean tratadas por simulación o por medios experimentales. En ambos casos, se busca predecir ese desempeño y garantizar eficiencia en las aplicaciones. Para los ingenieros y especialistas en computación de alto desempeño esto deberá generar situaciones más transparentes a los usuarios científicos, que cada vez exigen una gran transferencia de datos y un tratamiento eficiente de la complejidad. Así mismo, los avances tecnológicos ofrecen nuevas posibilidades de transferencia: redes que garantizan una alta y altísima velocidad de transferencia y capacidades cada vez mayores, exigen la creación de protocolos de transferencia y gestión de datos que permitan la utilización de todos los recursos disponibles, en este contexto, compartidos y distribuidos. Dentro de las perspectivas de trabajo en esta área, están igualmente el diseño de herramientas de monitoreo tanto de infraestructuras, como de aplicaciones en ejecución y el diseño de gestores de recursos que dependiendo del desempeño de una aplicación pueda tratar situaciones críticas de transferencia, evitando por ejemplo, embotellamientos y bloqueos a nivel de red.

5. Conclusiones

En este artículo se trataron varios aspectos, que más que resolver preguntas, generarán preguntas para el tratamiento de esta problemática. Una de las preguntas que generará es como

garantizar una comunicación eficiente ante dos situaciones posibles en computación de alto desempeño: transferencia intensiva de datos y la transferencia de datos que representan mensajes de gran tamaño?. Diferentes estrategias que van desde el avance tecnológico de dispositivos hasta el diseño de algoritmos son propuestas por varios autores, sin embargo, se sugiere un mejor tratamiento del diseño del algoritmo teniendo en cuenta las posibilidades mínimas y el alcance máximo de la tecnología disponible. Claro, teniendo en cuenta las perspectivas de escalabilidad tanto de la arquitectura como de la aplicación misma. De ahí se deriva una segunda pregunta: Como predecir el comportamiento?. En este artículo se propone la observación directa por medio de la experimentación de la aplicación misma, pero fundamentada en un modelo matemático que permita su validación ... incluso con una comparación de los resultados simulados y los resultados reales.

El problema de la predicción es aún un problema abierto en muchos fenómenos. En este caso en particular, esa predicción no puede ser a través de un análisis aislado del fenómeno, sino también hay que tener en cuenta la interacción y el ambiente en la cual se está analizando el desempeño de la transferencia. Teniendo en cuenta esa complejidad derivada, varios autores proponen un análisis con modelos estocásticos, pero, en analogía con los fenómenos físicos y buscando la simplicidad en el modelamiento, es interesante estudiar otros métodos que pueden derivarse de modelos parametrizados. Obvio, como se observa en la dinámica que ha tenido el modelo LogP, pueden surgir nuevos parámetros para explicar comportamientos asociados y de la misma manera, pueden surgir anomalías y diferencias entre la medida experimental y la predicción dada por el modelo. El reto entonces sería más lograr explicar esas diferencias que reducirlas, para lograr una buena descripción de la transferencia intensiva y masiva de datos en infraestructuras de computación de alto desempeño.

6. Agradecimientos

Este artículo es fruto de algunas reflexiones en torno a la tesis doctoral realizada por el autor y agradece principalmente a los doctores Yves DENNEULIN⁸ y Michel RIVEILL⁹, directores de ese trabajo de investigación. Así mismo se agradece a los organizadores, participantes, patrocinadores y a todas las personas que hicieron posible la realización de la Conferencia Latinoamericana de Computación de Alto Desempeño 2007, pero muy especialmente al comité de la Universidad Industrial de Santander, en Bucaramanga, Colombia.

⁸Profesor del Instituto Nacional Politécnico de Grenoble - Laboratorio de Informática de Grenoble, Francia.

⁹Profesor de la Universidad de Nice en Sophia Antipolis - Laboratorio de Informática, Señales y Sistemas de Sophia Antipolis, Francia.

Referencias

- [1] A. Aggarwal, A. Chandra, and M. Snir, On Communication Latency in PRAM Computations, Proc. 1st ACM Symp. on Parallel Algorithms and Architectures, pp. 11-21, June 1989.
- [2] Alexandrov, A., Ionescu, M., Schauser, K. and Scheiman, C. *LogGP: Incorporating Long Messages into the LogP Model*. Proceedings in 7th Annual ACM Symposium on Parallel Algorithms and Architectures, Santa Barbara, CA, U.S.A. 1995.
- [3] Barrios-Hernandez, C. and Denneulin, Y. *High Bandwidth Data Transfer Analysis and Modeling in Grids*. Proceedings in EXPEGRID Workshop in 15th HPDC 2006. Paris, France, 2006.
- [4] G. Bilardi, K.T. Herley, A. Pietracaprina G. Pucci, and P. Spirakis *BSP vs LogP*. Proceedings of 8th ACM Symposium on Parallel Algorithms and Architectures, pages 25–32, Padova, Italy, June 1996
- [5] Casanova, H. *Network Modeling Issues for Grid Application Scheduling*. San Diego Computer Center, Department of Computer Science and Engineering, University of California at San Diego, U.S.A., 2003.
- [6] Culler, D., Karp, Patterson, D., Sahay, A., Schauser, K., Santos, E., Subramonian, R., Von Eicken, T. *LogP: Towards a Realistic Model of Parallel Computation*. Proceedings in Four ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, San Diego, C.A. U.S.A. 1993.
- [7] Feng, W. and Tinnakornsriruphap, P. *The Failure of TCP in High-Performance Computational Grids*. Proceedings of the 2000 ACM/IEEE conference on Supercomputing. Dallas, Texas, U.S.A. 2000.
- [8] Foster, I. et Kesselman, C. *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann Publishers, U.S.A. 1998.
- [9] Frank, M., Agarwal, A. and Vernon, M. *LoPC: Modeling Contention in Parallel Algorithms*, Proceedings in 6 ACM SIGPLAN Symposium in Principles and Practice of Parallel Programming, Las Vegas, Nevada, U.S.A. June 1997.
- [10] GGF Network Measurements Working Group. *A Hierarchy of Network Performance Characteristics for Grid Applications and Services*. Internet Document, <http://nmwg.internet2.edu/docs/nmwg-measurements-v14.pdf> . 2003.
- [11] Grid 5000. *Grid 5000 Project*. <http://www.grid5000.fr> France.
- [12] GridSim. *GridSim Project*. <http://www.gridbus.org/gridsim/> Australia.
- [13] Infiniband Trade Association. *Infiniband*. <http://www.infinibandta.org/>
- [13] ISO/IEC 7498-1. *Information Technology– Open Systems Interconnection– Basic Reference Model: The Basic Model*. Internet Document, <http://www.iso.org> .
- [14] Kielmann, T., Bal, H., et Verstoep, K. *Fast Measurement of LogP Parameters for Message Passing Platforms*. Lecture Notes In Computer Science; Vol. 1800, Proceedings of the 15th IPDPS 2000 Workshops on Parallel and Distributed Processing, Springer - Verlag, U.K., 2000.
- [15] Lai, K. et Baker, M. *Measuring Link Bandwidths Using a Deterministic Model of Packet Delay*. Proceedings in Special Interest Group on Data Communications, Stockholm, Sweden. 2000.
- [16] Marchal, L. *A Network Model for Simulation of GRID Application*. Stage Report of MIM-2. ENS-Lyon, France, 2002.
- [17] Moritz, C. et Frank, M. *LogGPC: Modeling Network Contention in Message-Passing Programs*. IEE Transactions on Parallel and Distributed Systems. Vol. 12. No. 4, U.S.A. 2001.
- [18] Myricom Inc. *Myrinet* <http://www.myri.com/>
- [19] Németh, Z. et Sunderam, V. *Characterizing Grids: Attributes, Definitions, and Formalisms*. Journal of Grid Computing 1; Kluwer Academic Publishers, Netherlands, 2003.
- [20] Ould-Khaoua, M., Sarbazi-Azad, H. et Obaidat M. *Performance Modeling and Evaluation of High Performance Parallel and Distributed Systems*. Performance Evaluation Journal No. 60, Editorial. U.S.A. 2004.
- [21] Schoonderwoerd, R. *Network Performance Measurement Tools: A Comprehensive Comparison*. Master Thesis, Vrije Universiteit, Netherlands, 2002.
- [22] SimGrid. *SimGrid Project*. <http://simgrid.gforge.inria.fr/> France.
- [23] Tracy, G. *LogGP: Analysis of Parallel Sorting Algorithms*. Internet Document. <http://www.cs.wisc.edu/gtracy/classes/747/project.htm> . University of Winsconsin. U.S.A. 2000.