

Cálculo en Tiempo Real de Identificadores Robustos para Objetos Multimedia Mediante una Arquitectura Paralela CPU-GPU

Natalia Miranda, Fabiana Piccoli

LIDIC- Univ. Nacional de San Luis

San Luis, Argentina

Edgar Chávez

Univ. Michoacana de San Nicolás de Hidalgo

Morelia, México

{ncmiran, mpiccoli}@unsl.edu.ar, elchavez@umich.com

1. Contexto

Esta propuesta de trabajo se lleva a cabo dentro de la línea de investigación “Procesamiento de Información Multimedia” del proyecto “Nuevas Tecnologías para el Tratamiento Integral de Datos Multimedia”. Dicho proyecto se desarrolla en el marco del Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC), de la Facultad de Ciencias Físico, Matemáticas y Naturales de la Universidad Nacional de San Luis.

2. Resumen

El concepto de búsqueda exacta es central para los repositorios de información o bases de datos “tradicionales”. Trasladar el mismo concepto a repositorios de señales, no es simple. Por la naturaleza de los datos multimedia, el objetivo será resolver computacionalmente la similitud perceptual entre objetos similares. Dos objetos iguales perceptualmente deberían recibir distancias pequeñas; mientras que dos objetos perceptualmente distintos deberían recibir distancias grandes. Para ello es necesario contar con una representación robusta de cada dato del repositorio y definir una eficiente recuperación de la información en tiempo real .

Esta propuesta de trabajo establece los lineamientos a seguir para la identificación de señales, en particular las señales de audio y la recuperación de la información en forma rápida y efectiva, a través de la aplicación de técnicas de computación de alto desempeño.

Palabras Claves: Procesamiento de señales,

Computación de alta performance, Huellas digitales robustas, Espacios métricos

3. Introducción

Las bases de datos y repositorios de información multimedia (audio, imagen, video y texto) no pueden ser trabajadas tan eficientemente como en las bases de datos tradicionales debido a que la información multimedia debe ser recuperada por similitud; mientras que en las bases de datos tradicionales esta búsqueda es exacta.

Cada registro en una base de datos tradicional tiene un dato (la clave del registro), el cual permite recuperar todo el registro. En las bases de datos full text, como es el caso de la recuperación de información, con cualquier porción significativa de un texto podemos recuperar el texto completo; así, cuando buscamos en internet utilizamos como clave de búsqueda aquellas palabras que distinguen al texto que intentamos recuperar. Visto de otra manera, con una parte significativa; pero arbitraria, del objeto almacenado, podemos recuperar el objeto completo. En una base de datos multimedia deberíamos ser capaces de realizar la misma operación: con una parte del objeto almacenado sería suficiente para recuperar a todo el objeto. Esto es particularmente difícil si consideramos que los objetos multimedia no tienen una coincidencia digital; comparados bit por bit, dos objetos considerados iguales perceptualmente pueden no coincidir digitalmente (a menos que uno sea una copia digital del otro, lo cual es un caso poco interesante).

Para poder tratar a los objetos multimedia, en

particular a las señales de audio, con la misma eficiencia con las que se tratan a los objetos de texto, es necesario obtener su representación, estable y persistente a las diversas degradaciones naturales de los objetos. Esta representación es llamada la huella digital de la señal.

Adicionalmente, existen situaciones en las que se producen de manera simultánea varias señales, las cuales deben ser procesadas simultáneamente para ser analizadas contra un repositorio de información previamente almacenado. Un ejemplo de ello son las transmisiones radiales de las decenas de emisoras existentes en una ciudad particular. Una CPU dedicada con varios núcleos puede calcular en tiempo real, sin problema, las características de tantas señales como unidades de procesamiento posea; pero difícilmente podría hacerlo para las decenas de señales radiales coexistentes en el ámbito de la ciudad. Una alternativa a la CPU con varios núcleos es tomar ventaja de las propiedades de las tecnologías dedicadas.

El poder computacional asociado a las tecnologías dedicadas a fines específicos, sus constantes avances y el bajo costo, han constituido una alternativa válida a las supercomputadoras paralelas. El ejemplo más popular de las tecnologías dedicadas son las GPU (Unidad de Procesamiento Gráfico)[2, 19]. Una tarjeta de video puede proporcionar mucho más poder de cómputo que la computadora huésped en algunas aplicaciones[18].

Las tarjetas de video (GPU) han evolucionado mucho más rápido que los microprocesadores de propósito general [16, 19], esto ha permitido desarrollar un modelo de supercómputo casero en donde con recursos más económicos, de los que se requieren para comprar un PC, es posible programar en un modelo de paralelismo masivo. Esto se cumple para usable para cierto tipo de problemas en donde la arquitectura de procesadores con varios núcleos con memoria compartida y soporte multihilos puede ser utilizada.

Existen alternativas para procesamiento en GPU, la más ampliamente utilizada es la tarjeta Nvidia, para la cual se ha desarrollado un kit de programación en C, con un modelo de comunicación de datos y de control de hilos proporcionado por un driver, el cual provee una interfaz CPU-GPU [15]. Este ambiente de desarrollo llamado Compute Unified Device Architecture (CUDA) ha sido diseñado para simplificar el trabajo de sincronización de hilos y la comunicación con

la GPU [10, 20], proponiendo de esta forma un modelo de programación.

Se puede definir a CUDA como un ambiente de programación que permite a los desarrolladores de software crear componentes del programa aislados, las cuales resuelven un problema sobre un dispositivo dedicado, GPU, aplicando paralelismo de datos masivo. CUDA define un modelo de programación, el cual facilita el desarrollo de aplicaciones sobre la GPU. Ésta es considerada un procesador de muchos núcleos soportando numerosos threads de gránulo fino [22], mostrando flexibilidad en la asignación de recursos locales (memoria o registro) a los threads. Es el programador quien decide la división de los recursos, permitiendo así realizar ajustes para obtener mejoras en la performance.

El sistema de computación consiste de un host, CPU tradicional, y una o más dispositivos de cómputo, los cuales son los co-procesadores masivamente paralelos. Cada co-procesador aplica el modelo Simple-Instrucción-Multiples-Datos (SPMD), todos las unidades de cómputo (thread) ejecutan el mismo código, no necesariamente sincronizados, sobre distintos datos. Los threads comparten el mismo espacio de memoria Global [2, 10, 17, 20, 21].

En las siguientes secciones se explicarán las líneas de investigación y desarrollo así como los resultados obtenidos y esperados.

4. Líneas de Investigación y Desarrollo

Hay dos grandes problemas a resolver para poder manejar los objetos multimedia tan eficientemente como se manejan las bases de datos full text, como serían los buscadores de internet. El primero es la eficacia, o el poder de *indentificar* a los objetos sin error o con un margen pequeño de error, bajo diversas perturbaciones y degradaciones; y el segundo sería la eficiencia o el poder hacerlo *rápidamente* y de manera escalable [3, 8].

Existen dos maneras de acelerar una solución de búsqueda de patrones, una de ellas es encontrar un método de representación que tenga un comportamiento semejante a las búsquedas exactas, que tome tiempo constante para satisfacer una petición [4, 7]; la otra es utilizar un modelo de cómputo que permita la paralelización masiva a un costo relativamente bajo [11, 23]. Más interesante es la combinación de ambas: Determinar la

huella digital de señales de audio y realizar consultas en grandes base de datos en tiempo cercano al real. En consecuencia, las líneas específicas a seguir son:

- *Identificación de Señales:* Un modelo estándar de búsqueda en bases de datos multimedia consiste en utilizar una medida de (dis)similitud entre los objetos almacenados. Esta medida de distancia entre objetos debería modelar esencialmente el comportamiento de una persona al comparar dos objetos de esa naturaleza. Dos objetos iguales perceptualmente deberían recibir distancias pequeñas; mientras que dos objetos perceptualmente distintos deberían recibir distancias grandes. Si la distancia está efectivamente diseñada, será posible localizar un objeto en la base de datos comparando el objeto muestra con todos los objetos de la base datos [8]. Esto nos daría un mecanismo eficaz de recuperación; sin embargo, este mecanismo, no escala cuando la base de datos crece [3].

La representación de los objetos de forma estable y persistente a diversas degradaciones, tanto naturales como ataques maliciosos se denomina huella digital de la señal. Idealmente la huella digital debe ser una invariante de la señal; aquellas características intrínsecas, no alteradas por su constante manipulación.

La determinación de una huella digital permitirá, por ejemplo identificar objetos de audio a partir de segmentos, detección de duplicaciones, plagios, rotulado automático (MP3 modernos), consulta por ejemplos y filtrados en redes p2p, entre otros.

En [4, 5, 6], se discute un método muy eficaz para identificar sin error secuencias de audio sujetas a degradaciones severas; el método de identificación es secuencial. La identificación implica determinar la representación de la señal y su búsqueda en una base de datos de audio. Utilizando computación de alta desempeño se lograron mejorar los resultados citados haciéndolos eficientes para su cálculo en tiempo real [12, 13, 14].

- *Recuperación de la Información:* Para hacer escalable una solución efectiva es necesario

diseñar un índice que permita acceder a las mismas soluciones sin contrastarlo con todos los elementos de la base de datos.

Entre los distintos índices se encuentran los índices de permutaciones [9], los cuales han demostrado ser muy escalables [1]. Este método propone cambiar la comparación directa del objeto de consulta con cada elemento en la base de datos, por comparar solamente la similitud de sus permutaciones. La permutación de un elemento se da respecto a X , un conjunto de objetos distinguidos de la base de datos llamados permutantes. Cada elemento de la base de datos se representará con una permutación respecto a X . Cada elemento tiene una única perspectiva de la permutación y la consulta es resuelta sólo con aquellos elementos de la base de datos que tienen una perspectiva similar. Todos los elementos similares tendrían índice de permutación similar. La definición de similitud entre los elementos es fundamental para obtener buenos resultados.

Esta propuesta de indexación resuelve una consulta en dos fases, la primera para ordenar la base de datos de acuerdo a la distancia de la permutación, escogiendo los N primeros elementos como candidatos. La segunda fase consiste en refinar dicha lista y seleccionar los k vecinos más cercanos (KNN) del objeto de consulta (Recién en el proceso de refinamiento se compara el elemento de consulta con los seleccionados de la base de datos). Lo mismo ocurre para los otros tipos de consultas en base de datos. Desarrollar un índice de permutación para señales de audio y la correspondiente resolución de consultas teniendo en cuenta la necesidad de resolverlo en tiempo real es uno de nuestros desafíos.

Por lo expuesto anteriormente, se considera importante que para señales de audio, la demodulación de la cantidad de información es una característica esencial de la señal, la cual es estable bajo severas degradaciones y puede ser calculada en tiempo real utilizando una arquitectura CPU-GPU para conjuntos del orden de decenas de señales de audio.

5. Resultados obtenidos / esperados

De las líneas de investigación a seguir, en el caso de la identificación de señales de audio, se han desarrollado algoritmos para la determinación de la huella digital aplicando técnicas de alta performance utilizando la arquitectura CPU-GPU, los resultados obtenidos fueron muy buenos [12, 13, 14].

Actualmente se está trabajando en la otra línea de investigación propuesta, particularmente en el diseño de un índice basado en permutantes para realizar consultas y obtener los resultados esperados rápidamente, cuidando no sólo de realizar una comparación con todos los elementos de la base de datos, sino también utilizando la CPU-GPU como modelo de computación de alta performance.

6. Formación de Recursos Humanos

Los resultados esperados respecto a la formación de recursos humanos son hasta el momento una tesis de doctorado en desarrollo y la finalización de beca de postgrado de tipo II otorgada por el Consejo Nacional de Investigaciones Científicas (CONICET), recientemente obtenida.

Referencias

- [1] Amato G., Savino P. - Approximate similarity search in metric spaces using inverted files - InfoScale 08: Proceedings of the 3rd international conference on Scalable information systems - ICST (Institute for Computer Sciences, Social Informatics and Telecommunications Engineering) - Brussels, Belgium 2008 - Pp 1:10
- [2] Buck, I. - GPU computing with NVIDIA CUDA - SIGGRAPH '07: ACM SIGGRAPH 2007 courses ACM, New York, NY, USA. 2007.
- [3] Bustos Cárdenas, B.E. - Index Structures for Similarity Search in Multimedia Databases - PhD thesis, Universitat Konstanz, Fachbereich Informatik, Germany - Octubre 2006.
- [4] Camarena Ibarrola, A., Chavez E. - A robust entropy-based audio fingerprint - IEEE International Conference on Multimedia and Expo 2006.
- [5] Camarena Ibarrola A., Chávez E. - On Musical Performances Identification Entropy and String Matching - MICAI 2006
- [6] Camarena Ibarrola J.A. - Análisis digital de la señal de voz - PhD thesis Borrador - Universidad Michoacana de San Nicolás de Hidalgo, México - Agosto 2007.
- [7] Camarena Ibarrola, A., Chavez, E. - Real time tracking of musical performances. - In MICAI, volume To appear - 2010.
- [8] Chávez E., Navarro G., Baeza Yates R.A., Marroquín J.L. - Searching in metric spaces - ACM Comput. Surv. Vol 33 N3 - Pp 273:321 - 2001.
- [9] Chávez E., Figueroa K., Navarro G. - Effective proximity retrieval by ordering permutations. - IEEE Transactions on Pattern Analysis and Machine Intelligence - Vol 30 N 9 - September 2008 - Pp 1647:1658.
- [10] Chen, W. , Hang, H. - H.264/AVC motion estimation implementation on Compute Unified Device Architecture (CUDA) - IEEE International Conference on Multimedia and Expo 2008 - Pp 697:700 - April 2008.
- [11] Grama A., Gupta A., Karypis G., Kumar V. - Introduction to Parallel Computing - Addison Wesley - 2003.
- [12] Miranda N., Piccoli F., Chávez E. - Using GPU to Speed Up the Process of Audio Identification - I2TS 2010 - 9th International Information and Telecommunication Technologies Symposium. IEEE - R10. ISBN 978-85-89264-11-2. - Rio de Janeiro, Brazil - December 2010.
- [13] Miranda N., Piccoli F., Chávez E. Camarena Ibarrola A. - Finding Audio Fingerprint Using GPU - IX Congreso Argentino de Mecánica Computacional - XXXI Congreso Ibérico Latinoamericano de Métodos Computacionales en Ingeniería (Mecom - Ciilamce 2010) - ISSN 1666-6070.- Pp 3107:3126 - Noviembre 2010 - Buenos Aires, Argentina.
- [14] Miranda N., Piccoli F., Chávez E., Camarena Ibarrola A. - Fast GPU Audio Identification - 16vo Congreso Argentino de Ciencias de la Computación (CACIC 2010) - ISBN 978-950-9474-49-9. - Pp 229:242 - Univ de Moron, Buenos Aires, Argentina - Octubre 2010.

- [15] Joselli, M., Zamith, M., Clua, E., Montenegro, A., Conci, A., Leal-Toledo, R. Valente, L., Feijo, B., dórnellas, M., Pozzer, C - Automatic Dynamic Task Distribution between CPU and GPU for Real-Time Systems - . 11th IEEE International Conference on Computational Science and Engineering, 2008 (CSE '08) - Pp 48:55 - July 2008.
- [16] Lieberman, M.D. and Sankaranarayanan, J. and Samet, H. - A Fast Similarity Join Algorithm Using Graphics Processing Units - ICDE 2008. IEEE 24th International Conference on Data Engineering 2008 - Pp 1111:1120 - April 2008.
- [17] Lindholm E., Nickolls J., Oberman S. and Montrym J. - NVIDIA Tesla: A Unified Graphics and Computing Architecture Â Micro, IEEE - vol 28 N 2- Pp 39:55 - MarchÂApril 2008.
- [18] Lloyd, D., Boyd, C., Govindaraju, N. - Fast computation of general Fourier Transforms on GPUS - IEEE International Conference on Multimedia and Expo - Pp 5:8 - April 2008.
- [19] Luebke, D., Humphreys, G. - How GPUs Work Computer - vol 40, N 2 - Pp 96:100 - ISSN 0018-9162 - Feb. 2007.
- [20] Luebke, D. - CUDA: Scalable parallel programming for high-performance scientific computing - 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro. ISBI 2008. Pp 836:838 - May 2008.
- [21] Ogata Y., Endo T., Maruyama N., Matsuoka S. - An efficient, model based CPU-GPU heterogeneous FFT library. - IEEE International Symposium on Parallel and Distributed Processing. IPDPS 2008 - Pp 1:10 - ISSN 1530-2075 Â April 2008.
- [22] S. Ryoo, C.I. Rodrigues, S.S. Baghsorkhi, S.S. Stone, D.B. Kirk, W.W. Hwu - Optimization principles and application performance evaluation of a multithreaded GPU using CUDA-Principles and Practice of Parallel Programming, Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming. ISBN:978-1-59593-795-7. Pp 73-82. USA. 2008.
- [23] Yang L., Guo M. - High Performance Computing: Paradigm and Infrastructure - WileyÂInterscience - 2006.