Ingeniería de Computadores en la era del *Big Data*: Computación de altas prestaciones en clasificación y optimización

Julio Ortega; Pedro Martín-Smith; Jesús González Peñalver; Miguel Damas
Departamento de Arquitectura y Tecnología de Computadores
E.T.S.I.I.T., Universidad de Granada
jortega@ugr.es

Resumen. Este artículo describe las necesidades que las aplicaciones de la ciencia de datos plantean a las arquitecturas de computador y las consecuencias que comportan en la enseñanza de asignaturas de este ámbito. Como ejemplo, de esta relación entre aplicaciones y arquitectura de computadores se describe la asignatura Computación de Altas Prestaciones en Clasificación y Optimización, impartida en el Máster en Ciencia de Datos e Ingeniería de Computadores de la Universidad de Granada.

Palabras clave: *Big Data*, Ciencia de Datos, Clasificación, Computadores de Altas Prestaciones, Ingeniería de Computadores, Optimización.

Abstract. This paper describes the requirements that present big data applications demand from computer architectures and their consequences in the teaching of subjects in this area. As an example of these relationships between applications and computer architectures, the subject High Performance Computing on Classification and Optimization included in the Master of Data Science and Computer Engineering of the University of Granada (Spain) is described.

Keywords: Big Data, Classification, Computer Engineering, Data Science, High Performance Computers, Optimization.

1 Introducción

La capacidad para generar y almacenar información ha permitido disponer de grandes cantidades de datos y plantear nuevas aplicaciones de gran interés socio-económico que ahora sí podrían abordarse. Extraer conocimiento de estos grandes volúmenes de información plantea retos importantes, no solo a las técnicas de minería de datos existentes, sino también a las plataformas de cómputo donde se ejecutan [1]. Así, para conseguir una implementación eficiente de las aplicaciones de minería de datos que definirán la denominada era del *Big Data*, las arquitecturas de computador deben

satisfacer ciertos requisitos que determinarán las características de los computadores por la relevancia de las aplicaciones de datos y, por tanto, deberían tenerse en cuenta en el planteamiento de estrategias docentes para diseñar el proceso de enseñanza-aprendizaje en muchas de las asignaturas relacionadas con la ingeniería de computadores.

El Máster de Ciencia de Datos e Ingeniería de Computadores que se imparte en la Universidad de Granada [2] pone de manifiesto esta necesidad de tener en cuenta los requisitos de las aplicaciones de *Big Data* y las características de los computadores que permiten su implementación eficiente. En este artículo, además de detallar algunas de las implicaciones de la Ciencia de Datos en la Ingeniería de Computadores, se describe la asignatura de Computación de Altas Prestaciones en Clasificación y Optimización, como un ejemplo concreto con el que ilustrar esas implicaciones.

Así, la Sección 2 del artículo analiza las demandas que plantea la denominada Ciencia de Datos a la Ingeniería de Computadores, y las características tecnológicas que harán posible aplicaciones de los grandes volúmenes de datos existentes, y definirán el camino hacia la era del *Big Data*. La Sección 3 proporciona información sobre el Máster de Ciencia de Datos e Ingeniería de Computadores que se imparte en la Universidad de Granada. La asignatura Computación de Altas Prestaciones en Clasificación y Optimización se encuentra incluida en la especialidad de Ingeniería de Computadores y Redes del máster mencionado, y se analiza con detalle en la Sección 4. Finalmente, las conclusiones del artículo se recogen en la Sección 5 y las referencias se indican en la Sección 6.

2 Computadores para el *Big Data*

En el informe “Frontiers in Massive Data Analysis" del Nacional Research Council de Estados Unidos [1] se describen, entre otros aspectos los límites para el desarrollo de las aplicaciones de análisis de grandes volúmenes datos. Los datos generados en experimentos, simulaciones, o recogidos por sensores dan lugar a bases de datos con tamaños del orden de los petabytes. En 2010 se generaron 1.2 Zettabytes (10^{21} bytes, es decir, mil trillones de bytes), Facebook genera alrededor de 10 Tbytes/día, y Twitter, 7 TB/día. Compañías como Google o Microsoft disponen de datos en cantidades del orden del exabyte (10^{18} bytes), es decir, trillones de bytes (hay que tener en cuenta que desde el Big Bang se estima que han pasado 10^{17} segundos). Se estima que el volumen de datos almacenado crece a un ritmo mayor que el que marca la ley de Moore para el número de transistores en un circuito integrado (que sirve también como ritmo de referencia para el ritmo de mejora de la capacidad de los computadores). Así, frente al factor de crecimiento de 2 cada dos años que parece indicar la ley de Moore actualmente, el volumen de datos almacenados parece presentar un factor de crecimiento próximo a 2.5 cada dos años. Esta diferencia en los ritmos de crecimiento, denominada brecha de los datos (*data deluge gap*) plantea retos importantes en cuanto a la captura, procesamiento, distribución, comunicación, y análisis de los datos. Por tanto, esta situación también tiene consecuencias importantes en el desarrollo de arquitecturas de computación eficientes [3].

En líneas generales, se puede decir que el almacenamiento, acceso y procesamiento de grandes volúmenes de datos requiere de sistemas de cómputo paralelos y distribuidos. De hecho, ya desde hace algún tiempo, los microprocesadores incorporan varios núcleos de procesamiento para poder mantener el ritmo de mejora de prestaciones que marca la ley de Moore, con las restricciones en la frecuencia de reloj que plantean las limitaciones en el consumo energético del microprocesador. Pero la capacidad de procesamiento constituye solo una parte de lo que deben ofrecer las arquitecturas. Las E/S y el almacenamiento también deben ser capaces de proporcionar acceso rápido a los datos. Los datos deben estar próximos a los procesadores de las plataformas paralelas/distribuidas que los procesan, y para conseguir un ancho de banda de E/S aceptable, también el almacenamiento (y los sistemas de ficheros) deben encontrarse distribuidos. La jerarquía de memoria también es esencial. Cada vez es más frecuente el uso de aceleradores que cooperan con las CPUs de los nodos. Entre estos aceleradores, las GPU (Unidades de Procesamiento Gráfico) se vienen utilizando frecuentemente y constituyen una alternativa muy popular de cara a conseguir ganancias de velocidad considerables con consumos muy competitivos. La comunicación a través de la memoria con estos aceleradores, y el uso de los distintos niveles de almacenamiento de las tarjetas de GPU también son elementos a tener en cuenta.

Además de su conocida ley sobre el paralelismo, Gene Amdahl también propuso una serie de leyes, o más bien reglas prácticas (denominadas en inglés *rules of thumb*) para el diseño de computadores [4,5]: la ley del sistema equilibrado (un sistema necesita un bit de E/S por segundo por instrucción por segundo), la ley de la memoria (en un sistema equilibrado la relación α=Bytes/IPS es igual a 1), y la ley de las E/S (los programas hacen una E/S por cada 50.000 instrucciones). Estas *leyes* pueden seguir teniéndose en cuenta actualmente, aunque con algunas modificaciones en los valores de algunos de los parámetros [5]. Así en [6] y [7] se analizan desde ese punto de vista los requisitos que plantean, respectivamente, los centros de datos, y los entornos de programación de aplicaciones de Big Data como Hadoop y DataMPI. En [7] se habla de la denominada segunda ley de Amdahl (que incluye las leyes del sistema equilibrado y de la memoria: por cada instrucción por segundo de velocidad de CPU se debe proporcionar un bit/s de E/S y un Byte de memoria principal) y, para evaluar los sistemas, se utilizan el número de memoria de Amdahl, o relación α=GBytes/GIPS (GIPS es Giga Instrucciones por segundo), y el número de E/S de Amdahl, NESA=Ancho de Banda (en bits/s)/IPS, que debería ser igual a 1 en un sistema equilibrado. Mientras que en sistemas de cómputo de altas prestaciones, para reducir el coste del sistema, se tienen números de Amdahl (valor mínimo de α y NESA) tan bajos como 0.001, en sistemas diseñados para el procesamiento de datos intensivo (DataScope, GrayWulf, sistema de análisis de datos de eBay) se tienen valores entre 0.5 y 1 [7].

Las tareas de adquisición, búsqueda o visualización tienen una complejidad que crece linealmente con el volumen de los datos. Sin embargo las tareas de minería de datos suelen tener complejidades no lineales, y por ejemplo, muchos algoritmos de clustering pueden tener una complejidad superior a N^2, siendo N el número de datos sobre los que se aplica: un incremento de mil en el tamaño de los ficheros de datos supone un aumento en el tiempo de procesamiento de un factor de más de un millón

[8]. Por otra parte, el carácter inherentemente distribuido de las aplicaciones implica una clara dependencia de la capacidad de las redes que interconectan los elementos del sistema. Esta dependencia no sólo se da a nivel del sistema de E/S sino también en la jerarquía de memoria de forma que, a medida que aumenta la utilización del procesador, también lo hace la de la memoria y las E/S [6]. Además, en la arquitectura de capas de un sistema de Big Data hay que tener en cuenta [7] tanto la capa de hardware (por ejemplo un cluster de computadores), la de software de sistema (sistema operativo, middleware de gestión del servidor, etc.), la del entorno para Big Data (Hadoop, DataMPI, etc.) y la de la aplicación propiamente dicha (aplicaciones de ordenación, generación de histogramas de datos, clustering, etc.). Por ejemplo, en el conjunto de benchmarks *BigDataBench* [9] se encuentran cargas de trabajo que presentan perfiles diferentes. Así, para MapReduce, *WordCount* (cuenta las veces que aparece una palabra en el conjunto de datos de entrada) implica una carga intensiva en el uso del procesador, *Sort* (ordena los datos de entrada en función de unas claves dadas) es intensiva en cuanto a E/S, y *K-means* (algoritmo que agrupa los datos en K *clusters* o agrupaciones) también es intensivo en cuanto al uso de CPU, y además ejecuta varias iteraciones de forma que el resultado de una iteración es la entrada de la siguiente, permitiendo analizar el uso que se hace de la memoria.

El análisis de las ejecuciones de los benchmarks con distintas aplicaciones y cargas de trabajo considerando su efecto en los tiempos de ejecución, fallos de cache y TLBs, tasa de error en las predicciones de los saltos, junto con los anchos de banda de E/S, de comunicación, uso de memoria y de CPU permiten analizar hasta qué punto una determinada configuración de aplicación/entorno de Big Data/SO/hardware es más o menos eficiente. Por ejemplo, en [7] se obtiene, para un conjunto de benchmarks con distintas cargas de trabajo (volúmenes de datos) y sobre una misma plataforma SO/hardware, que Hadoop es más lento que DataMPI, con números de Amdahl más pequeños. Con ello se tiene que DataMPI, al permitir mejores prestaciones con menos exigencia en cuanto a ancho de banda de E/S y memoria, es más eficiente que Hadoop. En [10] se utilizan los números de Amdahl para evaluar distintos sistemas de bases de datos para distintas cargas de trabajo y se proponen una serie de conclusiones a tener en cuenta en el diseño de una plataforma para Big Data, poniendo de manifiesto el gran potencial para el desarrollo de nuevas arquitecturas orientadas al Big Data. En general, y dada la dinámica y diversidad (tanto dentro de una aplicación como entre distintas aplicaciones) en cuanto a patrones de uso y requisitos de los recursos, se promueven los sistemas heterogéneos con subsistemas equilibrados entre sí, y optimizados para las distintas fases de una carga de trabajo, y para las cargas de trabajo más representativas.

3 El Máster de Ciencia de Datos e Ingeniería de Computadores

El máster de Ciencia de Datos e Ingeniería de Computadores de la Universidad de Granada es impartido por profesores de los Departamentos de Arquitectura y Tecnología de Computadores y Ciencias de la Computación e Inteligencia Artificial desde el curso 2014/15. En la página web del máster [2] se puede encontrar información sobre los objetivos y las competencias, el plan de estudios, y el resto de

datos académicos del título. Tal y como se recoge en dicha página web, a través de este máster se busca la sinergia entre ámbitos de investigación que abarcan desde los niveles del desarrollo de algoritmos inteligentes hasta su implementación en distintas plataformas hardware, con el denominador común de las aplicaciones que implican el análisis de grandes volúmenes de datos. Teniendo en cuenta la tendencia actual hacia sistemas que interconectan una gran cantidad de dispositivos y sensores y a la disponibilidad de infraestructuras con grandes volúmenes de almacenamiento, es previsible la demanda de especialistas competentes en los contenidos del máster. El máster está estructurado en dos especialidades (especialidad en Ingeniería de Computadores y Redes, y especialidad en Ciencia de Datos y Tecnologías Inteligentes) que se corresponden, respectivamente, con los niveles más relacionados con el desarrollo de algoritmos y con la configuración de plataformas físicas, aunque incluyendo asignaturas que contemplan la interrelación entre las características del hardware y los requisitos de los algoritmos para una ejecución eficiente.

Tabla 1. Estructura de módulos y créditos del Máster en Ciencia de Datos e Ingeniería de Computadores de la Universidad de Granada [2]

<table>
<tr><td>Módulo Obligatorio
(12 créditos)</td><td>Metodología de la Investigación
Introducción a la Ciencia de Datos
Emprendimiento y Transferencia del Conocimiento</td><td></td></tr>
<tr><td>Módulo de Nivelación de Conocimientos
(8 créditos)</td><td>Servidores Seguros
Sistemas empotrados y co-diseño hw/sw
Minería de datos: preprocesamiento y clasificación
Minería de datos: aprendizaje no supervisado y detección de anomalías</td><td></td></tr>
<tr><td rowspan="2">Especialidad
(24 créditos de especialidad elegida + 4 de cualquier especialidad</td><td>Especialidad de Ingeniería de Computadores y Redes</td><td>Módulo de Computación de Altas Prestaciones
Módulo de Sistemas de Aplicación Específica</td></tr>
<tr><td>Especialidad en Ciencia de Datos y Tecnologías Inteligentes</td><td>Módulo de Modelos Avanzados de Ciencia de Datos
Módulo de Big Data y Cloud Computing
Módulo de Tecnologías Inteligentes e Inteligencia Computacional
Módulo de Aplicaciones en Ciencia de Datos y Tecnologías Inteligentes</td></tr>
<tr><td>Proyecto de Fin de Máster
(12 créditos)</td><td colspan="2"></td></tr>
</table>

En la Tabla 1 se muestra la organización en módulos y los créditos que debe superar el estudiante en cada uno de ellos para alcanzar el título. Como se puede ver, se deben cursar 12 créditos de un bloque obligatorio que incluye asignaturas fundamentales de formación para el investigador en TIC (Metodología de la Investigación, Introducción a la Ciencia de Datos, y Emprendimiento y Transferencia del Conocimiento). Hay un bloque de nivelación de conocimientos que incluye cuatro asignaturas de las que el estudiante debe cursar dos (8 créditos) para completar su formación previa en los aspectos que sean precisos para la especialidad que vaya a cursar. Finalmente el estudiante debe cursar las asignaturas de la especialidad que haya elegido: un total de 24 créditos de asignaturas de dicha especialidad, más cuatro créditos que pueden ser de la propia especialidad o de la que no ha elegido. Los 60 créditos del máster se completan con el Proyecto de Fin de Máster de 12 créditos. Más detalles sobre este máster se pueden encontrar en [11].

La especialidad de Ingeniería de Computadores y Redes incluye dos módulos, el de Computación de Altas Prestaciones, y el de Sistemas de Aplicación Específica. La asignatura de Altas Prestaciones en Clasificación y Optimización se imparte dentro del módulo de Computación de Altas Prestaciones en la especialidad de Ingeniería de Computadores y, precisamente, se encuentra entre esas asignaturas, a las que nos referimos más arriba, que ilustran los beneficios que los distintos tipos de computadores paralelos pueden aportar a la ejecución eficiente de procedimientos recientemente propuestos para resolver problemas de clasificación y optimización complejos. En la siguiente sección se analizan sus contenidos.

4 Computación de Altas Prestaciones en Clasificación y Optimización

El modelo de mente basada en el reconocimiento de patrones pone de manifiesto la importancia de la clasificación en el ámbito de los sistemas cognitivos y en los proyectos relacionados con el estudio del cerebro y las teorías acerca de la mente. Por otra parte, muchas de las aplicaciones del aprendizaje automático (*machine learning*) y la inteligencia artificial implican problemas de clasificación y optimización costosos en cómputo y almacenamiento. En las aplicaciones de minería de datos se busca descubrir patrones potencialmente útiles en los conjuntos de datos de los que se dispone, y se plantea la construcción de modelos descriptivos y predictivos eficientes que se ajusten a los datos disponibles, tengan capacidad generalizadora y permitan decidir acerca de acciones futuras. Por tanto, la minería de datos implica resolver, entre otros, problemas de selección de características, clasificación, clustering, y optimización. Las aproximaciones con que se abordan estos problemas, y las arquitecturas de cómputo que hacen posible su ejecución, se ven afectadas por el incremento exponencial de los volúmenes de datos a procesar en las aplicaciones de Big Data [12]. La asignatura se centra en los problemas de clasificación, clustering, y optimización, que constituyen el núcleo de muchas de las aplicaciones de Big Data.

Los temas a través de los cuales se han organizado los contenidos a impartir son los siguientes:

1. Arquitecturas de computador paralelas
2. Introducción a los Modelos con paralelismo implícito en problemas de clasificación y optimización: Modelos bioinspirados (redes neuronales y computación evolutiva)
3. Clasificación y Clustering con Modelos Neuronales: Sistemas Autoorganizativos.
4. Computación evolutiva paralela en problemas de optimización mono y multi-objetivo.
5. Implementaciones en plataformas paralelas y distribuidas: arquitecturas multi-núcleo, multicomputadores, e infraestructuras de cloud.
6. Aplicaciones de los problemas de clasificación y optimización complejos: Big Data, BCI, bioinformática, detección de intrusos en redes, etc.

El primer bloque de contenidos incluye los conceptos relacionados con las arquitecturas de computador que pueden dar respuesta a las necesidades de las aplicaciones de clasificación y optimización. Se revisan los distintos tipos de arquitecturas paralelas, el tipo de paralelismo que implementan, y las tendencias futuras en las mismas, poniendo de manifiesto la interacción entre los requisitos que plantean las aplicaciones más demandadas en cada momento y las innovaciones que van incorporando las arquitecturas de los computadores. En esta línea, se estudiarán las arquitecturas multinúcleo, incluyendo las arquitecturas multinúcleo heterogéneas y las unidades de procesamiento gráfico (GPU, Graphic Processing Units) muy extendidas actualmente como coprocesadores para acelerar la ejecución de ciertas aplicaciones. También se ilustrará la importancia de la jerarquía de memoria a través de los distintos tipos de memorias existentes en arquitecturas multinúcleo como las GPU y los procesadores de red (NP), y no menos importante será el análisis de la gestión del almacenamiento de disco y las consecuencias que plantean en el mismo las aplicaciones que utilizan grandes volúmenes de datos.

Una vez descritas las plataformas, se pasa a estudiar los algoritmos que utilizarán esas arquitecturas para proporcionar el nivel de prestaciones requerido por las aplicaciones que los usan. Así, se consideran los modelos bioinspirados como son las redes neuronales artificiales y los algoritmos evolutivos. Aparte de analizar sus características y de ilustrar el tipo de aplicaciones en las que se utilizan, también se contempla su perfil desde el punto de vista del paralelismo implícito que poseen, y de su implementación en plataformas paralelas y distribuidas de altas prestaciones, para abordar problemas de clasificación y optimización complejos que aparecen en aplicaciones de interfaz cerebro-computador (BCI, *Brain-Computer Interface*), bioinformática, detección de intrusos en redes, etc. Tras estudiar características de estos modelos bioinspirados, entre los que están la autoorganización, se analizan con detalle los mapas autoorganizativos (SOM) [13] y su uso en problemas de clasificación y clustering, para pasar a la computación evolutiva en problemas mono y multiobjetivo. A continuación se considerarán los aspectos relacionados con su implementación eficiente en distintos tipo de plataformas (entre ellas las de tipo

cloud), los paradigmas de programación y las técnicas con buenas características de elasticidad y disponibilidad (entre ellas MapReduce).

Un ejemplo de uso de arquitecturas de altas prestaciones en una aplicación que incluye optimización y clasificación, y que ilustra la perspectiva desde la que se plantea la asignatura, se puede encontrar en las tareas propias de las BCI basadas en la clasificación de electroencefalogramas (EEG). Por un lado se utilizan bases de datos correspondientes a electroencefalogramas de distintos sujetos y estados mentales a clasificar. Estos ficheros pueden tener un tamaño considerable al incluir señales recogidas a través de varios electrodos, y dadas sus características de baja relación señal-ruido de las señales, y el carácter temporal y no estacionario de las mismas. Requieren un preprocesamiento relativamente elaborado, y la caracterización de las mismas para su clasificación no es trivial. Así, usualmente, el número de características o componentes (*features* en inglés) que describen un EEG suele ser muy elevado, y puesto que, por otra parte, el número de EEG de que se dispone no es demasiado grande (dado el relativo coste temporal que implica la obtención de EEGs), el problema de clasificación de EEGs suele presentar el denominado problema de la maldición de la dimensionalidad (*curse of dimensionality*, en inglés). Es necesario llevar a cabo una selección de las características esenciales que permitan disponer de una relación suficientemente grande entre patrones (EEGs correspondientes a distintos estados mentales) y componentes de dichos patrones (características de los EEGs).

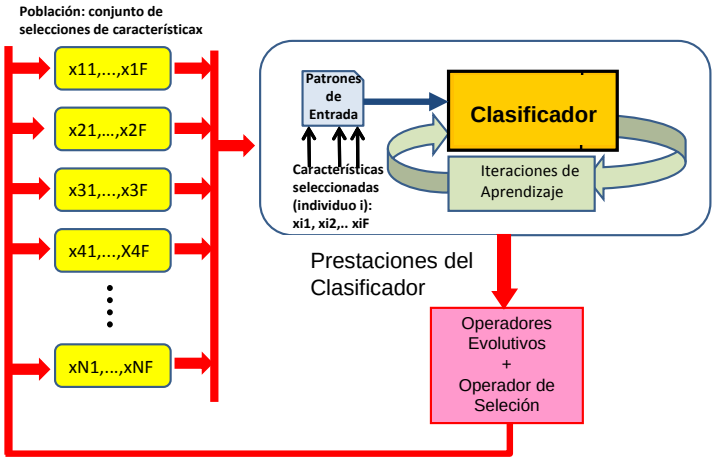

Figura 1. Procedimiento de tipo *wrapper* para la selección mediante optimización

En la Figura 1 se representa un procedimiento que permite la selección de características mediante un algoritmo de optimización evolutivo que tiene en cuenta las prestaciones que alcanza un clasificador entrenado (ajustado) con patrones que tienen como componentes las características cuya selección está codificada a través de los individuos de la población del algoritmo evolutivo. En este procedimiento de tipo *wrapper*, la calidad de los individuos de la población del algoritmo evolutivo viene dada por las prestaciones alcanzadas por el clasificador utilizado, y estas prestaciones

pueden definirse no sólo a través de un índice como puede ser el índice de aciertos de clasificación para los patrones utilizados como conjunto de test, sino también evaluar sus propiedades de generalización. Así, se puede introducir el uso de la optimización multi-objetivo en la selección de características. Igualmente, en el caso de que no se disponga de patrones etiquetados con sus clases, la evaluación de las características seleccionadas se puede hacer a partir de un procedimiento de *clustering* no supervisado, y se pueden estudiar el uso de mapas autoorganizativos, por ejemplo.

En cuanto al uso de arquitecturas paralelas, dado que el número de características entre las que hay que encontrar la mejor selección es muy elevado (cientos de características), estamos ante un problema de alta dimensionalidad en el espacio de variables de decisión del algoritmo de optimización multiobjetivo. En esta situación, el uso de computadores paralelos es esencial. En la asignatura se analizarán los distintos tipos de algoritmos paralelos (evaluación paralela del fitness de los individuos de la población, evaluación de subpoblaciones, etc.) y sus implementaciones (maestro-trabajador, islas, etc.), y se proporcionarán modelos de prestaciones que permitan entender las condiciones en las que una alternativa puede ser más beneficiosa que otra. Una alternativa para el aprovechamiento del paralelismo de datos que está recibiendo mucha atención es el uso de las GPU (*Graphic Processing Units*). La Figura 2 proporciona un esquema que ilustra las ventajas y los problemas de la implementación de un algoritmo evolutivo (en este caso un algoritmo de optimización multiobjetivo basado en el NSGA-II) con paralelismo basado en evolución de subpoblaciones en una arquitectura de GPU. Si bien es posible aprovechar el paralelismo de datos al evaluar en paralelo los individuos de la población, hay que tener en cuenta que deben implementarse operadores de selección, determinación de relaciones de dominancia, operadores de cruce, etc., que implican la comparación de varios individuos, y por lo tanto cierta comunicación y sincronización entre hebras. Por otra parte, si la evaluación de las soluciones implica ajustar un clasificador, o un mapa autoorganizativo, utilizando información de ficheros de patrones de un tamaño superior al de la memoria local disponible, el costo de las transferencias de memoria para acceder a dicha información (conexiones de los bloques de *fitness evaluation* a la memoria global de la Figura 2) puede comprometer el beneficio que aporta el aprovechamiento del paralelismo de la arquitectura. En cualquier caso, se trata de una oportunidad única para poner de manifiesto la interacción entre procesadores, jerarquía de memoria, y sistemas de interconexión, y analizar el compromiso entre paralelismo y localidad que exigen las aplicaciones consideradas y que determina la capacidad de una arquitectura dada para satisfacer los requisitos de prestaciones que se establecen.

En resumen, y teniendo en cuenta estos planteamientos, a través del temario de la asignatura se pretende que el estudiante sea capaz de:

- Identificar el paralelismo implícito en los modelos bioinspirados neuronales y evolutivos y las posibilidades de las arquitecturas de cómputo paralelas actuales que pueden aprovechar.
- Proponer modelos neuronales o evolutivos plausibles en la resolución de problemas de clasificación y optimización.

- Identificar los principios del comportamiento autoorganizativo y aplicarlos en problemas de clustering y clasificación.
- Distinguir entre problemas de optimización mono y multi-objetivo y estimar las diferencias en complejidad que plantea su resolución mediante aproximaciones basadas en computación evolutiva paralela.
- Identificar y proponer distintas alternativas para la implementación paralela de procedimientos de clasificación y optimización teniendo en cuenta las características de las arquitecturas de cómputo a utilizar (multiprocesadores, multicomputadores, o plataformas distribuidas) y los paradigmas de programación (entre estos, paradigmas como MapReduce para el tratamiento de grandes volúmenes de datos).
- Proponer procedimientos de clasificación y optimización de altas prestaciones en ejemplos de aplicaciones de complejidad elevada o que impliquen un procesamiento de información no estructurada (análisis de datos complejos, Brain-Computer Interfaces, etc.).
- Implementar los procedimientos de clasificación y optimización estudiados a través de herramientas de programación (como por ejemplo Matlab u Octave).

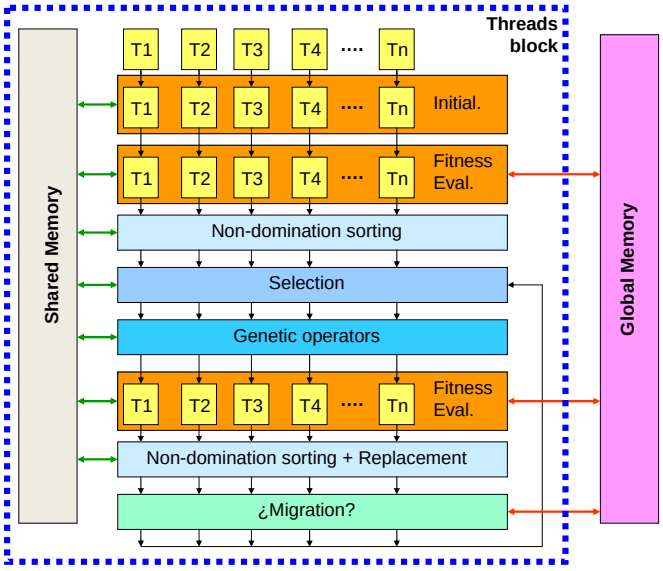

Figura 2. Esquema de implementación en una GPU de la evolución de una subpoblación en un procedimiento evolutivo de optimización multiobjetivo.

5 Conclusiones

La asignatura Computación de Altas Prestaciones en Clasificación y Optimización pone de manifiesto los aspectos de esas aplicaciones que requieren el uso de arquitecturas paralelas, y la evolución previsible en las arquitecturas de computador para dar respuesta a las exigencias de la minería de datos en el contexto del Big Data. Las competencias específicas del título de máster en Ciencia de Datos e Ingeniería de Computadores que el estudiante adquirirá a través de esta asignatura son las siguientes:

- Capacidad para la aplicación de técnicas y metodologías que permitan abordar desde nuevas perspectivas los problemas de interés, gracias a la disponibilidad de las plataformas de computación y comunicación con altos niveles de prestaciones.
- Capacidad de análisis de aplicaciones en ámbitos de biomedicina y bioinformática, optimización y predicción, control avanzado, y robótica bioinspirada, tanto desde el punto de vista de los requisitos para una implementación eficaz de los algoritmos y las técnicas de computación que se usan para abordarlas, como de las características deseables en las arquitecturas donde se ejecutan.

6 Referencias

1. National Research Council. “Frontiers in Massive Data Analysis”. Washington, D.C.: The National Academies Press, 2013.
2. Página web del Máster en Ciencia de Datos e Ingeniería de Computadores (Universidad de Granada). http://masteres.ugr.es/datcom/pages/master, 2015.
3. HiPEAC (European Network of Excellence on High Performance and Embedded Architecture and Compilation): “On the road for the HiPEAC Vision 2015”. https://www.hipeac.org/publications/vision/
4. Amdahl, G.M.:”Computer Architecture and Amdahl’s Law”. IEEE Computer, pp.38-46. Diciembre, 2013.
5. Gray, J.; Shenoy P.: “Rules of Thumb in Data Engineering”. In Proc. of the 16th IEEE Int. Conf. on Data Engineering, pp.3-12, 2000.
6. Cohen, D.; Petrini, F.; Day, M.D.; Ben-Yehuda, M.; Hunter, S.W.; Cummings, U.:”Applying Amdahl’s other law to the data center”. IBM J. Research & Development, Vol.53, No. 5, pp.683-694, 2009.
7. Liang, F.; Feng, C.; Lu, X.; Xu, Z.:”Performance characterization of Hadoop and DataMPI based on Amdahl’s second law”. 19th IEEE Int. Conf. on Networking, Architecture and Storage, pp.207-215, 2014.
8. Bell, G.; Gray, J.; Szalay, A.: “Petascale Computational Systems: Balanced CyberInfrastructure in Data-Centric World”. 2005.
9. Wang, L.; et al.:”BigDataBench: a Big Data Benchmark suite from Internet services”. Proc. of the 20th IEEE Int. Symp. On High Performance Computer Architecture (HPCA’14), 2014.
10. Chang, J.; Lim, K.T.; Byrne, J.; Ramirez, L.; Ranganathan, P.: “Workload diversity and dynamics in Big data analytics: Implications to System Designers”. In Proc. 2nd Workshop on Architectures and Systems for Big Data (ASBD’12), 2012.
11. Rojas, F.; Cano, A.; Gómez, M.; Ortega, J.; Herrera, F.; Romero-Zaliz, Rocío, González, J.:”Máster en Ciencia de Datos e Ingeniería de Computadores: una apuesta por la formación especializada en el sector TIC”. Revista de Enseñanza y Aprendizaje de Ingeniería de Computadores, 2015.
12. Wu, X.; Zhu, X.; Wu, G.-Q.; Ding, W.:”Data Mining with Big Data”. IEEE Trans. On Knowledge and Data Engineering, Vol.26, No.1, pp.97-107, 2014.
13. Kohonen, T.:”Self-Organizing Maps”. Springer, 2001.