Computación de Altas Prestaciones sobre Entornos Grid en la Simulación Eléctrica de Tejidos Cardiacos

J.M. Alonso[1], J.M. Ferrero (Jr.)[2], V. Hernández[1], G. Moltó[1], M. Monserrat[2] y J. Saiz[2]

Resumen— **La simulación de la propagación del potencial de acción en tejidos cardiacos representa una tarea computacionalmente intensa, con elevados requerimientos de memoria. El uso de detallados modelos celulares iónicos, combinado con su aplicación en geometrías tridimensionales, convierten a dicha simulación en un problema únicamente abordable, en tiempo razonable, mediante el empleo de Computación de Altas Prestaciones. Este artículo presenta la integración de un sistema paralelo para la simulación de la propagación del potencial de acción en tejidos cardiacos en un sistema de computación en Grid. La combinación de ambas técnicas computacionales ha permitido una aceleración en la ejecución de casos de estudio cardiacos mediante la realización de múltiples ejecuciones paralelas de manera concurrente en los recursos computacionales distribuidos de un Grid.**

Palabras clave— **Tejidos Cardiacos, Computación en Grid, Computación de Altas Prestaciones**

I. Introducción

LAS arritmias cardiacas son una de las primeras causas de mortalidad en los países desarrollados. Entre ellas, las taquicardias ventriculares y la fibrilación ventricular destacan debido a su capacidad de provocar la muerte súbita. A pesar de la intensa investigación, el mecanismo de la generación, desarrollo y terminación de estas arrítmias no se comprende todavía claramente.

Recientemente, los modelos matemáticos de la propagación de la actividad eléctrica han sido considerados como una herrramienta potente y útil para comprender mejor los mecanismos involucrados en el desarrollo de las arrítmias ventriculares.

La actividad eléctrica de las células cardiacas se describe por medio de modelos que detallan los movimientos de iones a través de la membrana celular. Combinando la formulación matemática del movimiento de los iones con la complejidad estructural del corazón es posible simular mediante un computador la propagación eléctrica en los tejidos cardíacos.

No obstante, la simulación de la propagación del potencial de acción representa un problema computacional complejo. Los pequeños pasos de tiempo empleados en la discretización temporal necesaria para la resolución de la ecuación (1), la cual gobierna este fenómeno en un modelo cardiaco monodominio, así como el elevado número de células presentes y los requerimientos de memoria, implican que el problema precise de técnicas de Computación de Altas Prestaciones para su resolución.

Esto resulta especialmente importante en un tejido tridimensional, donde una simulación del potencial de acción durante unos pocos milisegundos puede llegar a durar varios días en una plataforma secuencial.

$$\nabla \cdot \sigma \nabla Vm = C_m \cdot \frac{dVm}{dt} + I_{ion} + I_{st}. \qquad (1)$$

La ecuación anterior relaciona el potencial de membrana de las células, Vm, la suma de las corrientes iónicas que atraviesan la membrana, I_{ion}, la capacidad de membrana, C_m, el tensor de anisotropía, σ y el estímulo eléctrico, I_{st}. Para el cálculo del término I_{ion} se ha empleado el complejo modelo celular Luo-Rudy Fase II [1], el cual describe con detalle el funcionamiento interno de la célula cardiaca.

Además del inherente coste computacional de una simulación, existen multitud de estudios que requieren la ejecución de una gran cantidad de simulaciones paramétricas. Por ejemplo, probar los efectos de un nuevo fármaco requiere la ejecución de multiples simulaciones donde se varía la concentración de la droga. Además, para estudiar los efectos de la isquemia tardía es necesario variar las resistencias de acoplamiento entre las células en un determinado intervalo y observar la evolución de la actividad eléctrica del tejido bajo diferentes condiciones de anisotropía.

Por lo tanto, para poder abordar toda esta complejidad computacional se ha optado por integrar dos tecnologías diferentes. En primer lugar, la Computación de Altas Prestaciones permite reducir el tiempo de ejecución de una única tarea, simulando tejidos de mayor dimensión durante más tiempo, mediante la ejecución en un cluster de PCs. Por otra parte la computación en Grid aparece como una solución para el uso colaborativo de recursos computacionales entre diferentes organizaciones [2].

De esta manera, la combinación de ambas estrategias surge como la solución más apropiada para la aceleración de la ejecución de casos de estudio cardiacos. En el presente trabajo se describirá la integración de un sistema paralelo, desarrollado para la simulación de la propagación del potencial de acción de tejidos cardiacos, en una infraestructura Grid.

El artículo está estructurado como sigue. En primer lugar, la sección II describe las principales

[1]Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. Camino de Vera s/n, 46022, Valencia. España. E-mail: jmalonso,vhernand,gmolto@dsic.upv.es. [2]Departamento de Ingeniería Electrónica. Universidad Politécnica de Valencia. Camino de Vera s/n, 46022, Valencia. España. E-mail: cferrero,monserr,jsaiz@eln.upv.es.

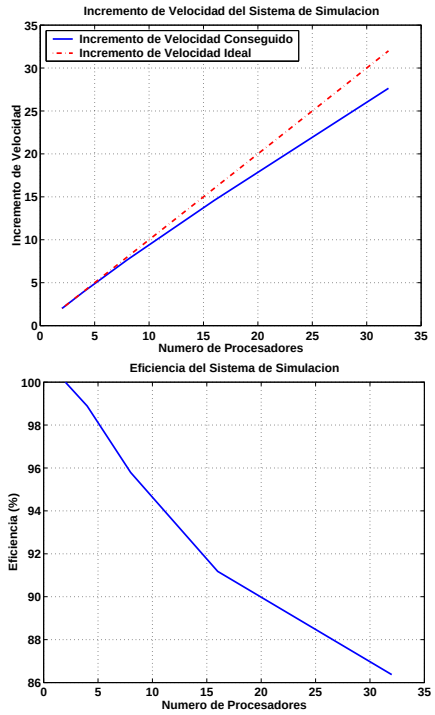


Fig. 1. Incremento de velocidad y eficiencia del sistema de simulación. Ejecución en un cluster de PCs con 20 biprocesadores Pentium Xeon a 2.0 Ghz, con 1 GByte de RAM y una red SCI.

características del sistema de simulación, destacando las modificaciones que se han tenido que realizar para permitir su ejecución en Grid. Posteriormente, la sección III describe el sistema de computación en Grid desarrollado bajo el middleware Globus Toolkit. La sección IV presenta un desarrollo alternativo realizado con el programa comercial InnerGrid. A continuación, la sección V analiza un caso de estudio cardiaco para evaluar los beneficios del uso de las tecnologías Grid y, finalmente, la sección VI presenta las conclusiones destacando los resultados obtenidos.

II. Características del sistema de simulación

La simulación de la actividad eléctrica en tejidos cardiacos consiste en un proceso iterativo que permite calcular la evolución del potencial de membrana de las células que lo componen a lo largo de un periodo de tiempo. En nuestro caso, en primer lugar se desarrolló un sistema de simulación paralelo para tejidos bidimensionales con el objetivo de reducir el tiempo de simulación en un cluster de PCs con buenos resultados de eficiencia [3], [4].

Posteriormente se mejoró el sistema para realizar simulaciones tridimensionales del tejido cardiaco. La Figura 1 muestra el incremento de velocidad y la eficiencia obtenido en una simulación de propagación de potencial de acción durante 250 ms en un tejido de 100x100x100 células, con un paso de tiempo de 10 μs, sobre un cluster de PCs. Dicha simulación requiere 177.97 horas en una plataforma secuencial, pero únicamente 6.45 horas si empleamos 32 procesadores.

El sistema de simulación periódicamente genera unos ficheros de copia de seguridad, independientes del número de procesadores, que permite retomar una simulación fallida a partir del instante en el que se salvó el estado. Para ello se emplean las rutinas de E/S paralela que ofrece la implementación de ROMIO [5] del estandar MPI-2 I/O.

A. Portabilidad e Interoperabilidad

Dado que el objetivo principal consiste en realizar la ejecución de las simulaciones en las máquinas que forman parte del Grid, hay que tener en cuenta el elevado grado de diversidad de las mismas. A priori, un Grid es una colección de recursos heterogéneos y, por lo tanto, se debe de alcanzar cierto nivel de portabilidad en el fichero ejecutable para poder conseguir una ejecución satisfactoria.

Para alcanzar este objetivo se deben evitar todo tipo de optimizaciones dependientes de la arquitectura, como las opciones de compilación `-march` o `-mcpu`. Además, las librerías numericas optimizadas dependientes de la plataforma, como las implementaciones de BLAS [6] y LAPACK [7], proporcionadas por la Intel Math Kernel Library, no deben ser empleadas, porque pueden dar lugar a la ejecución de instrucciones ilegales en la máquina remota si ambas arquitecturas no coinciden. Afortunadamente, las opciones tradicionales de optimización como por ejemplo `-O3` pueden ser usadas sin riesgo, ya que únicamente realizan procesos de reordenación de código y demás tareas independientes de la arquitectura.

Nuestro sistema ha sido enlazado estáticamente para generar un simulador autocontenido. La librería de paso de mensajes MPI también forma parte del ejecutable, enlazando estáticamente con una implementación de MPICH [8], [9] especialmente configurada para deshabilitar la comunicación mediante memoria compartida entre procesos del mismo nodo de un cluster, ya que es conocido que pueden haber problemas en la liberación de memoria debido al empleo de las rutinas de IPC de Unix System V [9].

Este simulador paralelo autocontenido puede ejecutarse en un amplio abanico de máquinas Linux, aislando, en la medida de lo posible, a la aplicación del entorno de ejecución. Esto ha sido comprobado mediante ejecuciones paralela en diferentes arquitecturas como Pentium III, Pentium Xeon, incluso en un Intel Itanium 2, con diferentes versiones de Linux, como Red Hat Linux Advanced Server, Fedora Core 1 y Debian GNU/Linux.

Hay que destacar que la aplicación funciona en modo compatibilidad en las plataformas Intel Itanium 2 (64 bits), aunque se ejecuta hasta 8 veces más lento que en un Intel Pentium Xeon (32 bits), plataforma en la cual se había compilado inicialmente. Por lo tanto, hemos compilado de forma nativa en la plataforma Itanium 2 para poder conseguir tiempos de ejecución comparables en ambas arquitecturas, y ser capaces de explotar las prestaciones de los nodos Itanium del Grid.

De esta manera, disponemos dos simuladores au-

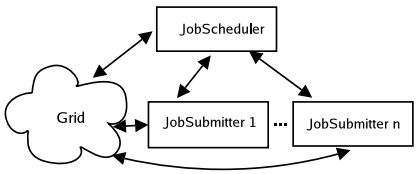


Fig. 2. Esquema del sistema de Computación en Grid desarrollado bajo Globus Toolkit.

tocontenidos, uno para las plataformas IA-32 y otro para IA-64, una estrategia que puede refinarse perfectamente para abarcar más arquitecturas.

III. Sistema de computación en grid bajo globus

En primer lugar, ha sido escogido el middleware Globus Toolkit [10] como infraestructura básica para el desarrollo del sistema Grid. Globus Toolkit es una iniciativa de código abierto que permite la construcción de Grids entre diferentes organizaciones, siendo, actualmente, el estándar de facto para la Computación en Grid.

La Figura 2 muestra una visión conceptual del sistema de computación en Grid desarrollado. El módulo `JobScheduler` realiza la asignación de simulaciones a recursos computacionales, delegando en una instancia del módulo `JobSubmitter` para cada simulación, el cual se encarga de la correcta ejecución de una tarea en el recurso escogido.

A. El Módulo JobScheduler

Este módulo toma como entrada un fichero con la descripción paramétrica de las múltiples simulaciones que forman un caso de estudio cardiaco. Para cada simulación, proporciona el mejor recurso disponible a partir de una lista predefinida de máquinas, consultando el número de procesadores libres del recurso, a través del servicio de Monitoring and Discovery Service (MDS) de Globus. Los clusters que tienen instalado el componente Globus Resource Allocation Manager (GRAM) Reporter indican el número de procesadores libres, delegando en el gestor de colas local (LoadLeveler, PBS, etc.).

En las estaciones de trabajo se obtiene una estimación del uso del procesador durante el último minuto para decidir si un recurso está disponible para aceptar una simulación o no.

Se han incluido en el sistema Grid prestaciones básicas de calidad de servicio, permitiendo especificar el número mínimo y máximo de procesadores que se deben asignar a una ejecución paralela. El número mínimo permite requerir cierta calidad de servicio, mientras que el número máximo permite, por un lado, aumentar la productividad mediante la ejecución de múltiples simulaciones en el recurso y, por otro lado, implementar una política respetuosa que no acapare todos los procesadores del recurso remoto.

Además, se ha incorporado un estimador a-priori, dependiente del problema, de la memoria necesaria para cada ejecución, lo cual previene el envío de simulaciones a máquinas que no disponen de la suficiente memoria.

Posteriormente, este módulo selecciona un ejecutable apropiado basado en la arquitectura de la máquina remota. El módulo JobScheduler también es responsable de lanzar las simulaciones pendientes, así como retomar las simulaciones fallidas, delegando para ello en una instancia del módulo JobSubmitter.

Si en un momento dado no existe ningún recurso libre disponible, periodicamente se encarga de comprobar la disponibilidad para poder lanzar las posibles simulaciones pendientes.

B. El módulo JobSubmitter

Cada instancia de este módulo se encarga de la correcta ejecución de una simulación en el recurso remoto. Para ello, en primer lugar, todos los ficheros de entrada que requiere el simulador son enviados a la máquina remota mediante el empleo del servicio GridFTP. A través de la interfaz de Globus se averigua la disponibilidad de la máquina remota para la ejecución de programas paralelos, de manera que se puede activar o desactivar la ejecución paralela automáticamente. La ejecución de la simulación se integra, si así ha sido configurado Globus, en el gestor de colas local de la máquina remota, respetando de esta manera, las políticas de ejecución de tareas de la organización.

Mientras el sistema de simulación se ejecuta en el recurso remoto, periodicamente genera unos ficheros de copia de seguridad para evitar comenzar una simulación desde el inicio en caso de fallo. Por lo tanto, este módulo envía periódicamente un trabajo a la máquina remota, el cual transfiere una imagen comprimida de los ficheros de copia de seguridad a la máquina local. De esta manera, los últimos ficheros de copia de seguridad siempre residen en la máquina local y una simulación fallida puede ser automáticamente relanzada en otro recurso computacional. Un mecanismo basado en firmas asegura que no se transfieran los mismos ficheros de copia de seguridad más de una vez, lo cual provocaría un desperdicio del ancho de banda.

Una vez que la ejecución ha finalizado, todos los ficheros de salida generados por el simulador se comprimen y son transferidos de nuevo a la máquina local, siendo guardados en una carpeta especial para la simulación. Asimismo, los ficheros temporales creados en la máquina remota son eliminados y, finalmente, el módulo JobSubmitter anota el estado de finalización de la tarea. Esta información será empleada por el módulo JobScheduler, que podrá conocer qué simulaciones han fallado y, de esta manera, relanzarlas de nuevo en un recurso disponible.

IV. Desarrollo de un sistema de computación en grid con el software innergrid

Además, hemos utilizado el programa comercial InnerGrid [11] como un middleware alternativo, de fácil utilización para la ejecución de trabajos

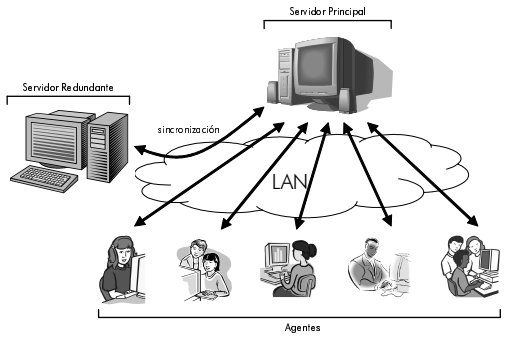


Fig. 3. Arquitectura de componentes en InnerGrid.

paramétricos secuenciales sobre un Grid formado por las máquinas de una organización. InnerGrid ofrece servicios de transferencia automática de los ficheros de los que depende la aplicación, combinado con un planificador de tareas propietario.

La Figura 3 muestra la arquitectura típica de componentes en InnerGrid. En todos los recursos que forman parte del Grid debe instalarse previamente un agente, el cual se conecta al servidor principal. Este servidor realiza las funciones de asignación de tareas y gestión del Grid, además de centralizar los ficheros de resultados generados por las ejecuciones.

Mediante una interfaz accesible via web, denominada InnerGrid Desktop, es posible tener una visión centralizada del Grid, consultando en cualquier momento el estado de las ejecuciones. De esta manera, se simplifica en gran medida el desarrollo necesario para poder ejecutar casos de estudio cardiacos.

En nuestro caso concreto, hemos desarrollado un nuevo módulo que permite especificar los requisitos tanto de memoria como de espacio en disco para ejecutar una simulación, y que define los parámetros que deben variar para el caso de estudio concreto. Posteriormente, se ha creado una tarea, que se puede ver como una instanciación del módulo, la cual permite especificar los rangos concretos de variación de los parámetros que intervienen en el estudio.

A continuación InnerGrid se encarga de realizar todas las funciones necesarias para la ejecución del conjunto de simulaciones resultantes en las máquinas disponibles del Grid. Resulta importante destacar que mediante la interfaz InnerGrid Desktop no es posible definir la recogida periódica de ficheros de copia de seguridad y, por lo tanto, la tolerancia a fallos se implementa por medio del reinicio de la simulación en un nodo libre del Grid.

V. Estudio de prestaciones

A. Infraestructura Grid

La Figura 4 muestra la infraestructura Grid de pruebas disponible, el cual está compuesto de recursos locales, pertenecientes a nuestro grupo de investigación, el Grupo de Redes y Computación de Altas Prestaciones de la Universidad Politécnica de Valencia (GRyCAP-UPV) y recursos remotos del Grupo de Arquitectura y Seguridad de Sistemas Distribuídos de la Universidad Complutense de Madrid

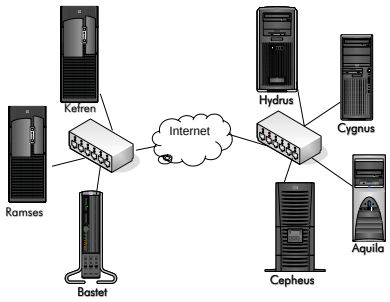


Fig. 4. Esquema de los recursos computacionales de la infraestructura Grid.

TABLA I
Características detalladas de las máquinas de la infraestructura Grid.

Nodo	Procesadores	Memoria
Kefren	10 (2 x Xeon 2.0 Ghz)	1 GB
Ramses	12 (2 x PIII 866 Mhz)	512 MB
Bastet	2 x Itanium II 900 Mhz	4 GB
Hydrus	1 x PIV 2.53 Ghz	512 MB
Cygnus	1 x PIV 2.53 Ghz	512 MB
Aquila	1 x PIII 666 Mhz	128 MB
Cepheus	1 x PIII 666 Mhz	256 MB

(ASDS-UCM). La Tabla I resume las principales características de las máquinas.

Globus Toolkit 2.4 está instalado en todas las máquinas del Grid, mientras que InnerGrid únicamente está disponible en los clusters Kefren y Ramses, puesto que no existe actualmente versión para Intel Itanium y dado que en las máquinas del ASDS-UCM no disponemos de potestad para la instalación de software.

B. Caso de Estudio: Isquemia de Miocardio

La isquemia de miocardio es una condición en la cual la ausencia de oxígeno en el músculo del corazón viene acompañada de una inadecuada retirada de los resultados del metabolismo debido a la reducción de flujo sanguíneo. Los mecanismos de la generación de taquicardias ventriculares y fibrilación ventricular (la más mortal de las arrítmias) pueden ser estudiados usando un modelo de tejido cardiaco regionalmente isquémico (resultante de la oclusión de una arteria coronaria) en el cual cierta parte del substrato es isquémico mientras que el resto del tejido adyacente permanece en condiciones normales [12].

En la zona afectada por el proceso de isquemia, los valores de varios parámetros electrofisiológicos sufren variaciones a lo largo del tiempo conforme la isquemia progresa. La concentración de potasio extracelular, en primer lugar, cambia desde su valor normal, 5.4 mmol/L, hasta un valor de 12.5 mmol/L durante los primeros 5 minutos de patología, alcanzando un estacionario durante los siguientes 5 minutos de isquemia [13]. En segundo lugar, la concentración intracelular de ATP decrece de manera lineal con el tiempo desde un valor normal de 6.8 mmol/L

a 4.6 mmol/L durante los primeros 10 minutos de isquemia, mientras que la concentración intracelular de ADP se incrementa de 15 μmol/L hasta 100 μmol/L en el mismo intervalo de tiempo [14]. Finalmente, la acidosis reduce la conductancia máxima de las corrientes de sodio y del calcio en un 25% de su valor normal durante los cinco y los diez minutos del episodio isquémico.

Por lo tanto, los cuatro parámetros mencionados cambian de forma diferente a lo largo del tiempo durante los diez primeros minutos de la isquemia de miocardio. En las simulaciones aquí presentadas, se ha simulado el comportamiento eléctrico a corto plazo del tejido en diferentes instantes de tiempo despues de la aparición de la situación de isquemia. El tiempo es, por lo tanto, el parámetro a variar en el conjunto de simulaciones.

El tejido 3D simulado comprende una zona central isquémica, formada por un cubo de 20x20x20 elementos (que representa un tejido de 2x2x2 mm) en el cual las condiciones de los parámetros mencionados varían con el tiempo, inmerso en un cubo de 60x60x60 elementos. Los parámetros electrofisiológicos del tejido que rodea a la zona central isquémica se mantienen en sus valores normales.

Este caso de estudio analizará la influencia en la propagación del potencial de acción de diferentes grados de condiciones de isquemia que ocurren entre entre los 0 y los 10 minutos de la aparición de la patología. Empleando un incremento de tiempo de 0.5 minutos, deberemos ejecutar 21 simulaciones paramétricas independientes que pueden ser ejecutadas en una infraestructura Grid.

Cada ejecución debe simular un periodo de 80 ms, empleando un paso de tiempo de 10 μs, aplicando un estímulo supra-umbral a todas las células del plano inferior del tejido durante el intervalo de simulación [50,52] ms. Cada 1 ms durante el intervalo [40,80] ms, se guardará una instantánea del valor del potencial de membrana de las células, dando lugar a un total de 68 MBytes de información. Además, cada simulación precisa de 180 MBytes de memoria RAM en una plataforma secuencial para poder ejecutarse.

C. Resultados de Ejecución

La Tabla II resume la distribución de tareas en el Grid empleando el sistema de computación basado en Globus. Como máximo, las ejecuciones paralelas se han limitado a un cuarto del total de procesadores disponibles en el recurso remoto. El número mínimo de procesadores se ajustó a uno, permitiendo de esta manera la ejecución secuencial de aplicaciones.

En dicha tabla, una entrada como 7 (8 p.) indica que se realizaron siete simulaciones con ocho procesadores cada una. Las máquinas Bastet y Cepheus no aparecen en la tabla porque estaban sobrecargadas durante la ejecución del caso de estudio y, por lo tanto, el planificador de tareas nunca las eligió. Además, la máquina Aquila no dispone de suficiente memoria RAM para albergar la simulación, motivo por el cual tampoco se le asignó ninguna tarea.

TABLA II
DISTRIBUCIÓN DE LAS SIMULACIONES EN LA INFRAESTRUCTURA GRID, PARA CADA MÁQUINA, EMPLEANDO EL SISTEMA DE COMPUTACIÓN BASADO EN GLOBUS TOOLKIT.

Machine	**Simulations**
Kefren	4 (4 p.), 3 (3 p.), 2 (2 p.) 2 (1 p.)
Ramses	1 (6 p.), 3 (5 p.), 1 (4 p.), 1 (1 p.)
Hydrus	2 (1 p.)
Cygnus	2 (1 p.)

La ejecución del caso de estudio completo duró 10.27 horas. Puede observarse como el planificador asigna dinámicamente a cada máquina un número de simulaciones de manera proporcional a su potencia computacional.

La ejecución del caso de estudio a través de InnerGrid necesitó 22.57 horas, distribuyendo las tareas y ejecutando secuencialmente entre los nodos de los clusters Kefren y Ramses.

La ejecución del mismo caso de estudio de una forma tradicional, es decir secuencialmente, en un nodo del cluster Kefren requirió 81.1 horas. Finalmente, ejecutando cada simulación con 4 procesadores en el mismo cluster (permitiendo de esta manera dos ejecuciones simultáneas), requirió un total de 11.9 horas.

Mientras que InnerGrid es más apropiado para aprovechar los ciclos libres de máquinas ociosas dentro de un Grid de una única organización, Globus Toolkit está centrado en recursos dedicados a través de Grids de diferentes organizaciones. Por lo tanto, un solución basada en Globus es mucho más apropiada para el problema de la simulación de la actividad eléctrica cardiaca, ya que permite la posibilidad de acceder a diferentes recursos computacionales de manera transparente.

D. Resultados del Caso de Estudio

La Figura 5 resume los resultados obtenidos mediante la ejecución del caso de estudio. En ella se representa el potential de membrana de las células de un corte vertical en el centro del cubo, para tres diferentes condiciones de isquemia, en cuatro instantes de tiempo distintos.

Las condiciones de isquemia a los 0 minutos no introducen ninguna interferencia en la propagación del potencial de acción, mientras que niveles más severos de isquemia provocan una ralentización en la propagación del potencial de acción dentro del area de tejido afectada. Este fenómeno se puede apreciar claramente mediante la representación tridimensional mostrada en la Figura 6, donde la propagación del potencial de acción avanza de abajo a arriba.

VI. CONCLUSIONES

En este artículo se ha presentado la integración de un sistema paralelo eficiente, para la simulación de la propagación del potencial de acción sobre un tejido cardíaco tridimensional monodominio, en un

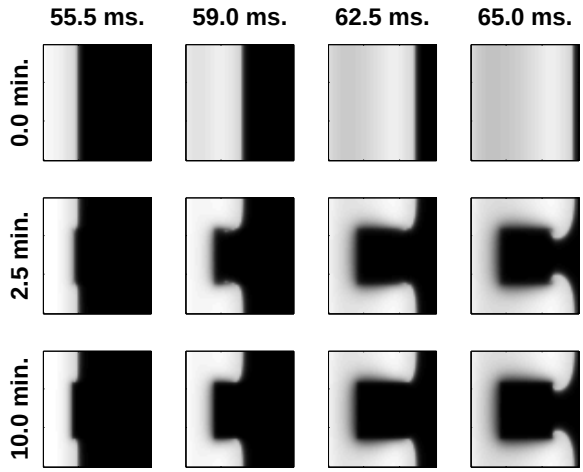
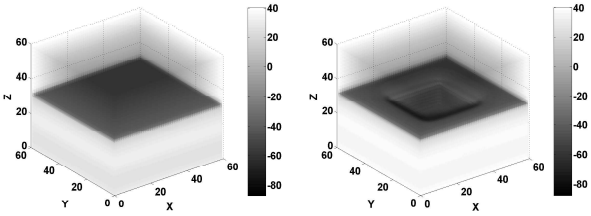


Fig. 5. Potencial de menbrana en cuatro instantes de tiempo bajo tres grados de isquemia. La propagación está aconteciendo de izquierda a derecha. La barra de colores es la misma que aparece en la Fig. 6.



(a) Isquemia tras 0 min. (b) Isquemia tras 10 min.

Fig. 6. Representación tridimensional del potencial de membrana de las células del tejido en el instante de simulación 59 ms.

Grid, con el objetivo de poder abarcar la demanda de recursos computacionales que requieren los casos de estudio cardiacos.

Para ello, se ha desarrollado un sistema de computación en Grid basado en Globus, que permite la integración de la Computación de Altas Prestaciones con las tecnologías Grid, mediante la ejecución paralela en máquinas multiprocesadores del Grid.

Además, se ha empleado el programa comercial InnerGrid como una alternativa de fácil utilización para la ejecución de aplicaciones paramétricas secuenciales en las máquinas de un Grid organizacional, desarrollando para ello un módulo específico para los casos de estudio cardiacos.

Así como la Computación de Altas Prestaciones es responsable del incremento de velocidad de una simulación, la computación basada en Grid es responsable de la productividad en la ejecución de múltiples simulaciones. Por ello, la integración de ambas estrategias computacionales ha resultado ser una combinación fundamental para aumentar la productividad en la ejecución de casos de estudio cardiacos.

Agradecimientos

Los autores desean agradecer el soporte financiero recibido de (1) el Ministerio de Ciencia y Tecnología para el desarrollo de los proyectos CAMAEC (TIC2001-2686) y GRID-IT (TIC2003-01318), cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través de los Fondos Estructurales, y (2) la Universidad Politécnica de Valencia por el proyecto CAMAV (20020418).

Además, agradecemos al Grupo de Arquitectura y Seguridad de Sistemas Distribuídos, del Departamento de Arquitectura de Computadores y Automática, de la Universidad Complutense de Madrid, por ofrecer tanto sus recursos de computación para aumentar nuestra infraestructura Grid como su conocimiento a través de los resultados obtenidos en el marco del proyecto GridWay [15].

Referencias

[1] C. H. Luo and Y. Rudy, "A Dynamic Model of the Cardiac Ventricular Action Potential. I Simulations of Ionic Currents and Concentration Changes," *Circulation Research*, vol. 74, pp. 1071–1096, 1994.

[2] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Lecture Notes in Computer Science*, 2001.

[3] J.M. Alonso, J.M. Ferrero (Jr.), V. Hernández, G. Moltó, M. Monserrat, and J. Saiz, "High Performance Cardiac Tissue Electrical Activity Simulation on a Parallel Environment," *Proceedings of the First European HealthGrid Conference*, pp. 84–91, January 2003.

[4] J.M. Alonso, J.M. Ferrero (Jr.), V. Hernández, G. Moltó, M. Monserrat, and J. Saiz, "Simulación de la Actividad Eléctrica Cardiaca: Una Implementación Basada en Computación de Altas Prestaciones," *Libro de Actas de las XIV Jornadas de Paralelismo, 15-17 de Septiembre de 2003, Leganés-Madrid*, pp. 33–38, 2003.

[5] Rajeev Thakur, William Gropp, and Ewing Lusk, "On Implementing MPI-IO Portably and with High Performance," *Proc. of the Sixth Workshop on I/O in Parallel and Distributed Systems*, pp. 23–32, May 1999.

[6] C. L. Lawson, R. J. Hanson, D. Kincaid, and F. T. Krogh, "Basic Linear Algebra Subprograms for FORTRAN usage," *ACM Trans. Math. Soft.*, vol. 5, pp. 308–323, 1979.

[7] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[8] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, "A High-Performance, Portable, Implementation of the MPI Message Passing Interface Standard," *Parallel Computing*, vol. 22, no. 6, pp. 789–828, 1996.

[9] William D. Gropp and Ewing Lusk, *User's Guide for MPICH, a Portable Implementation of MPI*, Mathematics and Computer Science Division, Argonne National Laboratory, 1996.

[10] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for the Distributed Systems Integration," *Infrastructure WG, Global Grid Forum*, 2002.

[11] GridSystems S.A., *InnerGrid Nitya Technical Specifications*, 2003.

[12] R. Coronel, "Heterogeneity in Extracellular Potassium Concentration During Early Myocardial Ischaemia and Reperfusion: Implications for Arrhytmogenesis," *Cardiovasc. Res.*, vol. 28, no. 6, pp. 770–777, 1994.

[13] J. N. Weiss, N. Venkatesh, and S. T. Lamp, "ATP-sensitive k^+ Channels and Cellular k^+ Loss in Hypoxic and Ischaemic Mammalian Ventricle," *J. Physiol.*, vol. 447, pp. 649–673, 1994.

[14] J. M. Ferrero (Jr.), J. Saiz, J. M. Ferrero, and N. Thakor, "Simulation of Action Potentials from Metabolically Impaired Cardiac Myocytes. Role of ATP-sensitive k^+ Current," *Circ. Res.*, vol. 79, pp. 208–221, 1996.

[15] E. Huedo, R. S. Montero, and I. M. Llorente, "A Framework for Adaptive Execution on Grids," *Software Practice and Experience*, To appear.