

Tesina de Licenciatura

Algoritmos paralelos sobre cluster de multicore. Aplicación a problemas de simulación

Silvana Lis Gallo

Licenciatura en Sistemas

**Director: Dr. Marcelo Naiouf
Codirector: Dra. Laura De Giusti**



Facultad de Informática
Universidad Nacional de La Plata

Contexto



III-LIDI



La presente tesina se encuentra enmarcada dentro de los proyectos del Programa de Incentivos del Ministerio de Educación:

- ✓ Arquitecturas multiprocesador distribuidas. Modelos, software de base y aplicaciones.
- ✓ Procesamiento paralelo y distribuido. Fundamentos y aplicaciones en sistemas inteligentes y tratamiento de imágenes y video.

Ambos, acreditados por la UNLP y desarrollados en el Instituto de Investigación en Informática III-LIDI.

Objetivos

- ✓ Investigación y desarrollo de algoritmos paralelos sobre multicores y cluster de multicores con aplicación a problemas con alta demanda computacional como es el caso de las simulaciones de eventos discretos.
- ✓ Estudio de los temas de investigación derivados de las técnicas de diseño y desarrollo de algoritmos y su evaluación en plataformas de memoria compartida, mensajes e híbridas con el objetivo de lograr eficiencia en el uso de los múltiples núcleos.
- ✓ Análisis de la posibilidad de aplicación de los resultados obtenidos en la solución de otros problemas de cómputo científico similares, desde el punto de vista de los requerimientos computacionales que plantea el caso de aplicación.

Agenda

- ✓ Conceptos de Sistemas Paralelos
- ✓ Conceptos de simulación
- ✓ Simulaciones paralelas y distribuidas
- ✓ Descripción del problema
- ✓ Solución secuencial
- ✓ Soluciones paralelas y distribuidas
- ✓ Pruebas Realizadas
- ✓ Conclusiones
- ✓ Trabajos Futuros

Agenda

- ✓ Conceptos de Sistemas Paralelos
- ✓ Conceptos de simulación
- ✓ Simulaciones paralelas y distribuidas
- ✓ Descripción del problema
- ✓ Solución secuencial
- ✓ Soluciones paralelas y distribuidas
- ✓ Pruebas Realizadas
- ✓ Conclusiones
- ✓ Trabajos Futuros

Conceptos de Sistemas Paralelos

Procesamiento Paralelo

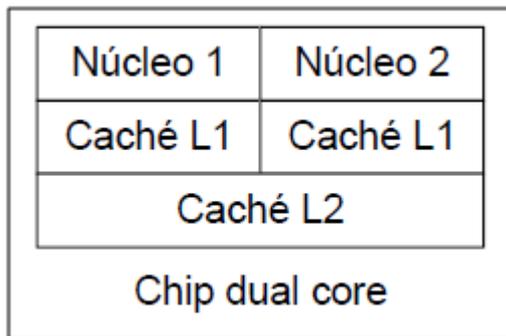
- ✓ El objetivo es resolver problemas más grandes, problemas con límite de tiempo y problemas con mayor precisión.
- ✓ Implica la utilización de programas con múltiples procesos o hilos, que se ejecutan en múltiples procesadores o núcleos.
- ✓ Posee múltiples aplicaciones.
- ✓ La tendencia ha sido hacia redes cada vez más veloces, sistemas distribuidos y arquitecturas con múltiples unidades de procesamiento, lo cual claramente demuestra que el futuro del procesamiento es el paralelismo.

Conceptos de Sistemas Paralelos

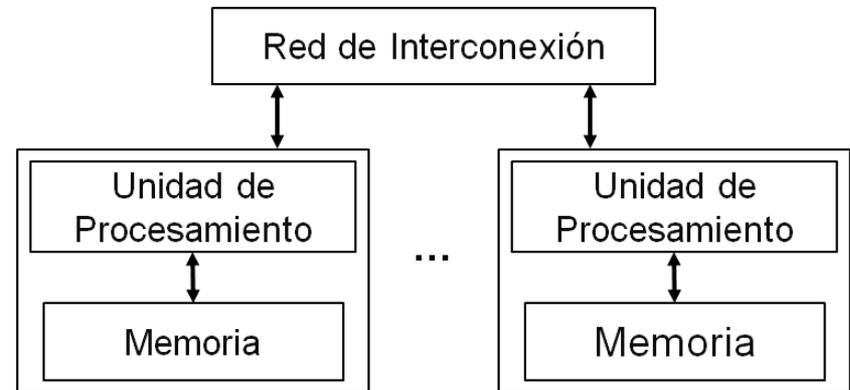
Arquitecturas

Una plataforma paralela consiste de dos o más unidades de procesamiento vinculadas mediante algún tipo de red de interconexión.

- ✓ Plataforma de **memoria compartida**: consiste de múltiples unidades de procesamiento conectadas a múltiples módulos de memoria.
- ✓ Plataforma de **memoria distribuida**: consta de p nodos de procesamiento unidos por una red de interconexión.



Multicore

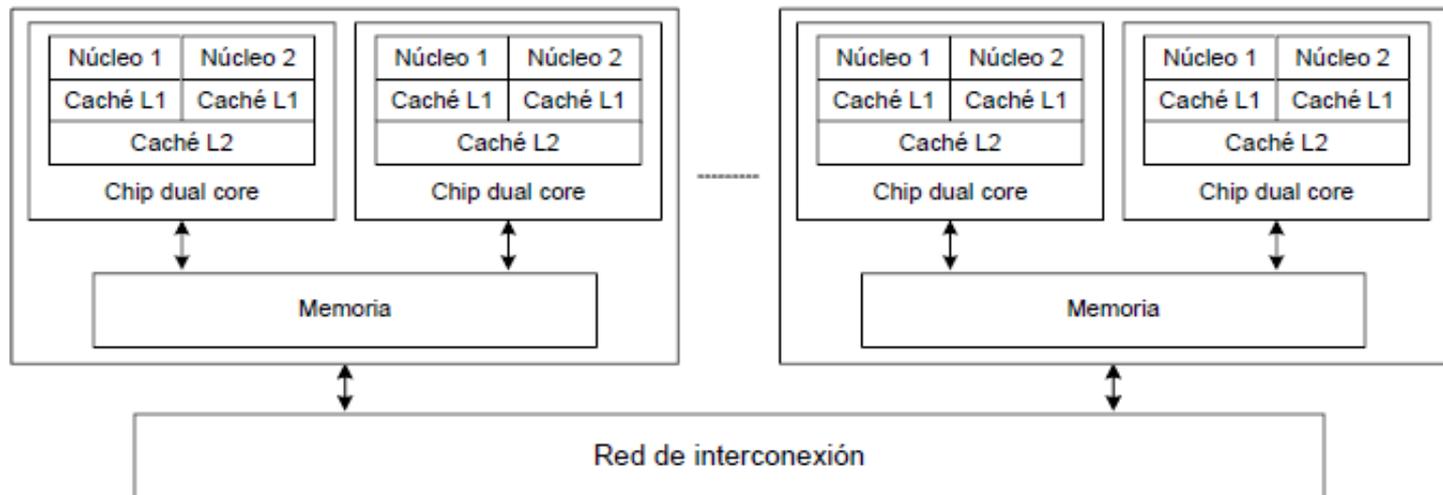


Cluster

Conceptos de Sistemas Paralelos

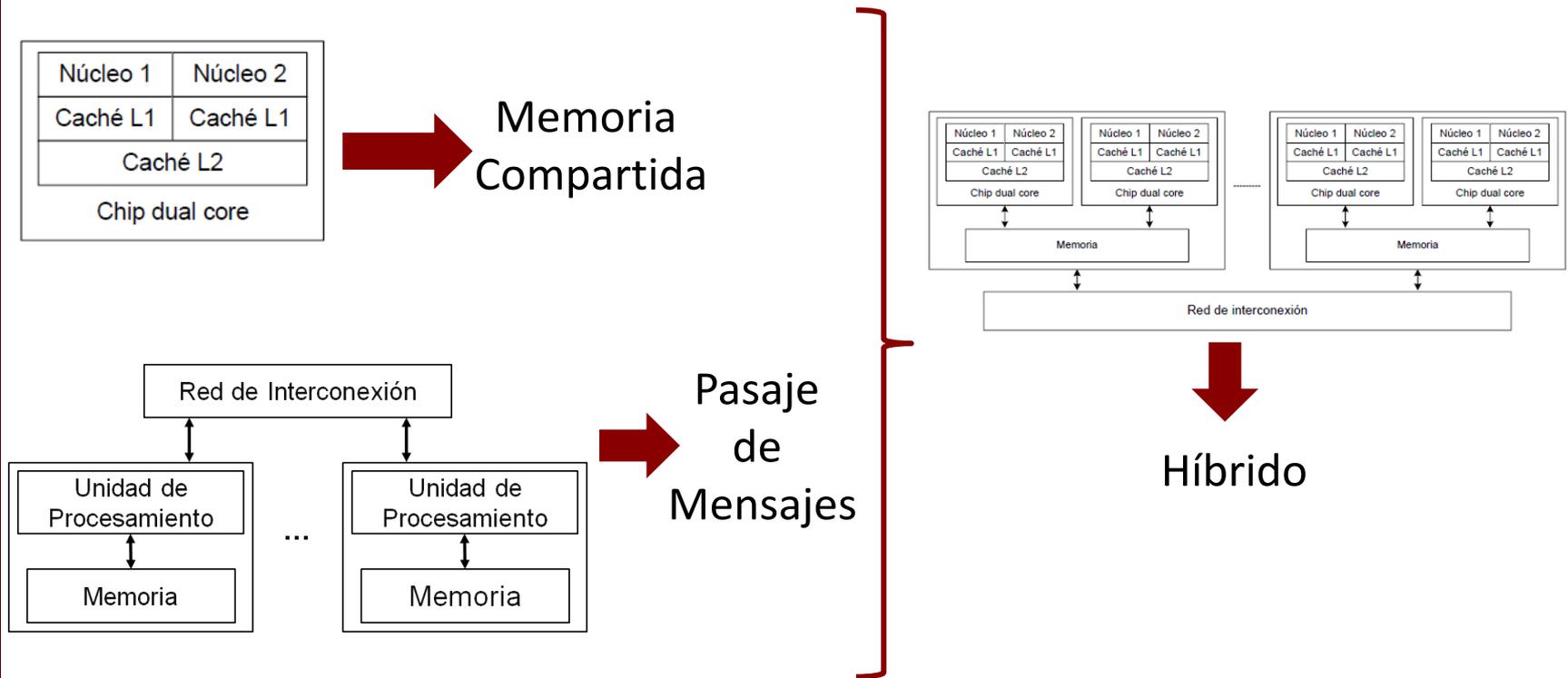
Cluster de multicore

- ✓ La aparición de la tecnología multicore ha impactado sobre los clusters.
- ✓ Los clusters de multicores agregan un nuevo nivel de comunicación.
- ✓ Estos cambios hacen necesaria la utilización de nuevas técnicas de programación paralela.



Conceptos de Sistemas Paralelos

Modelos de programación



Agenda

- ✓ Conceptos de Sistemas Paralelos
- ✓ **Conceptos de simulación**
- ✓ Simulaciones paralelas y distribuidas
- ✓ Descripción del problema
- ✓ Solución secuencial
- ✓ Soluciones paralelas y distribuidas
- ✓ Pruebas Realizadas
- ✓ Conclusiones
- ✓ Trabajos Futuros

Conceptos de Simulación

Modelos de simulación

✓ En la simulación por computadora, es necesario un **modelo** que represente el sistema, una especificación del **sistema físico** (o al menos de algunas de sus componentes) en términos de un conjunto de estados y eventos.

✓ Los modelos de simulación suelen clasificarse en:

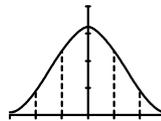
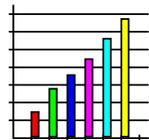
- Estáticos o dinámicos.



- Deterministas o estocásticos.



- Discretos o continuos.

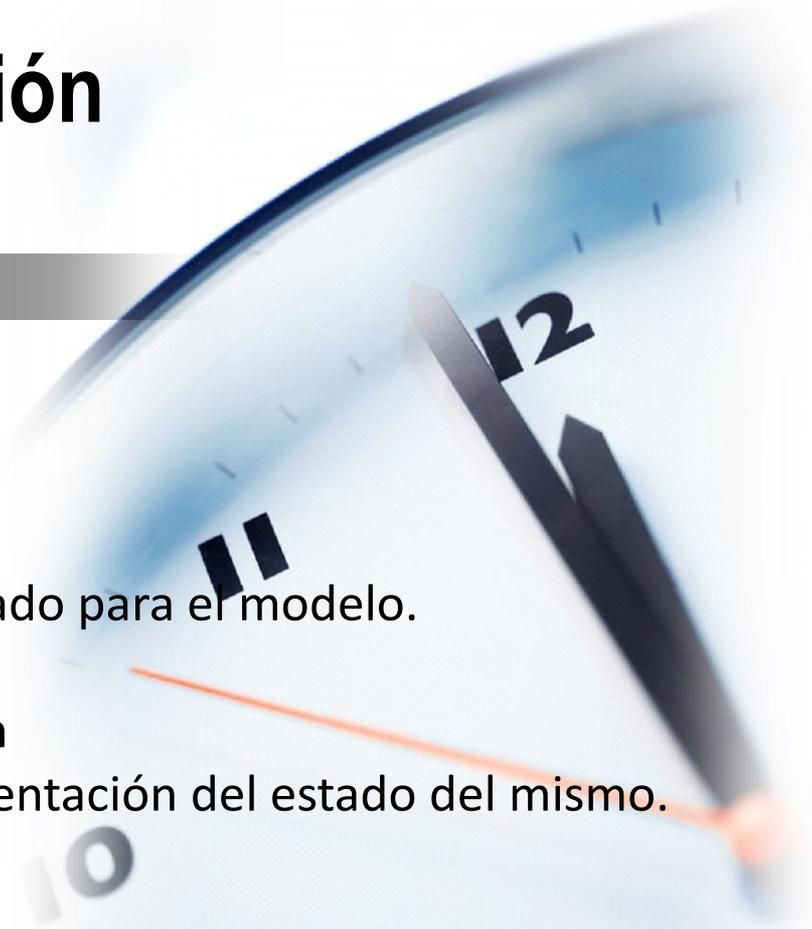


✓ La tesis se basa en los modelos de simulación **dinámicos**, **estocásticos** y **discretos**.

Conceptos de Simulación

Elementos

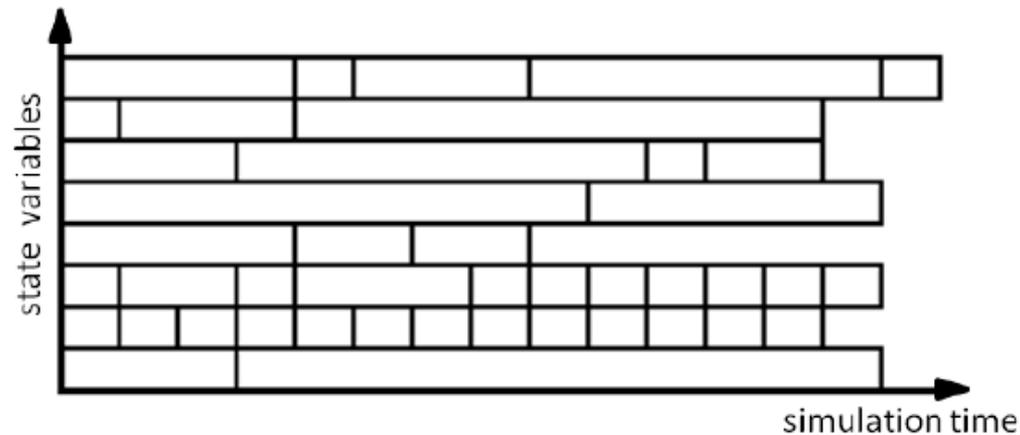
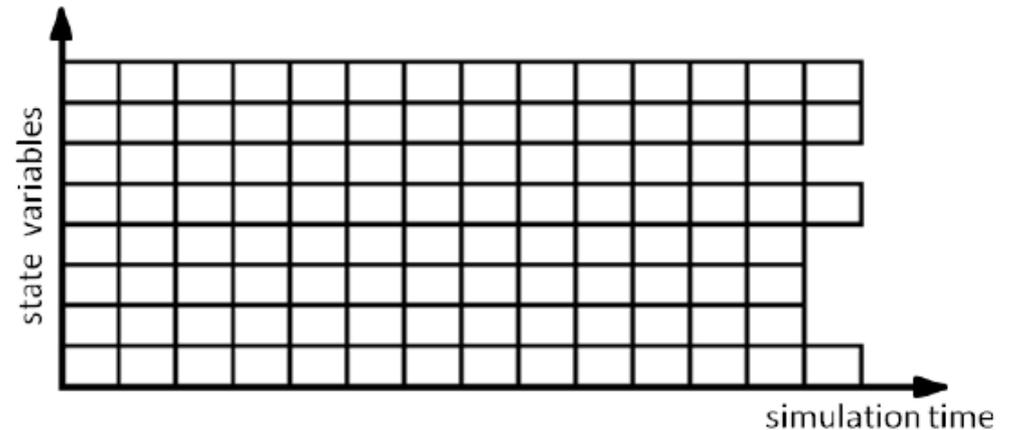
- ✓ Representación del **estado**
 - Variables de estado del sistema
 - EntidadesDe acuerdo al nivel de detalle deseado para el modelo.
- ✓ Mecanismo para modelar la **evolución**
 - Refleja sus resultados en la representación del estado del mismo.
- ✓ Representación del **tiempo**
 - Representa el avance del tiempo en el sistema físico.



Conceptos de Simulación

Formas de ejecución

- ✓ Ejecución time-stepped
- ✓ Ejecución event-driven



Agenda

- ✓ Conceptos de Sistemas Paralelos
- ✓ Conceptos de simulación
- ✓ **Simulaciones paralelas y distribuidas**
- ✓ Descripción del problema
- ✓ Solución secuencial
- ✓ Soluciones paralelas y distribuidas
- ✓ Pruebas Realizadas
- ✓ Conclusiones
- ✓ Trabajos Futuros

Simulaciones paralelas y distribuidas

Autómatas celulares

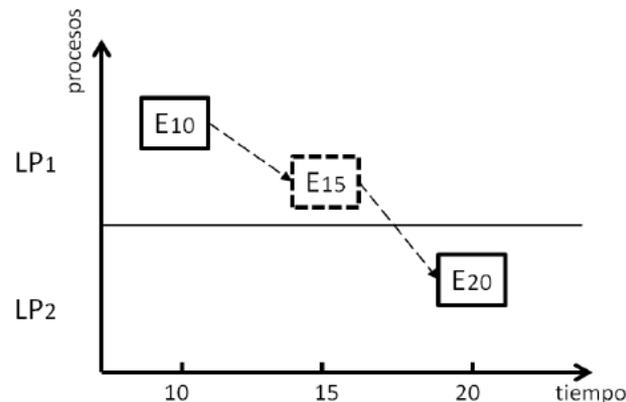
- ✓ Dentro de los enfoques de simulaciones existentes se puede incluir a los autómatas celulares.
- ✓ Consiste en una grilla o cuadrículado, donde cada celda de la cuadrícula se conoce como “célula”.
- ✓ Un conjunto finito de estados posibles.
- ✓ Cada célula tiene una “vecindad”.
- ✓ Una función de transición que calcula el nuevo estado que tendrá dicha cuadrícula en la siguiente etapa de tiempo.

Su importancia esta dada porque permiten obtener comportamientos complejos a partir de reglas muy sencillas.

Simulaciones paralelas y distribuidas

Sincronización conservativa y optimista

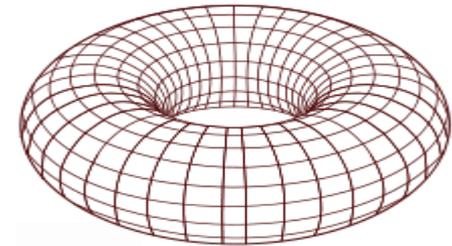
- ✓ Para la resolución de un modelo (cualquiera sea su enfoque), se utilizan procesos que ejecutan diferentes partes de la simulación.
- ✓ La sincronización utilizada entre dichos procesos puede realizarse:
 - Respetando la relación de causalidad de los eventos y ejecutando los eventos en orden → **Conservativa**
 - Permitiendo la ejecución desordenada de eventos → **Optimista**



Agenda

- ✓ Conceptos de Sistemas Paralelos
- ✓ Conceptos de simulación
- ✓ Simulaciones paralelas y distribuidas
- ✓ **Descripción del problema**
- ✓ Solución secuencial
- ✓ Soluciones paralelas y distribuidas
- ✓ Pruebas Realizadas
- ✓ Conclusiones
- ✓ Trabajos Futuros

Descripción del problema



- ✓ El problema consiste en la simulación de la evolución de una población dentro de un espacio representado mediante una grilla, siguiendo diferentes reglas.
- ✓ El lado derecho de la grilla se encuentra conectado con el izquierdo, y a su vez el lado superior con el inferior (es un espacio toroidal).
- ✓ Las reglas se encuentran definidas por el caso particular que se desea simular.
- ✓ En este caso, se tomó como ejemplo el problema de Tiburones y Peces.
- ✓ La simulación se lleva a cabo por una cantidad determinada de generaciones o pasos de tiempo.

Agenda

- ✓ Conceptos de Sistemas Paralelos
- ✓ Conceptos de simulación
- ✓ Simulaciones paralelas y distribuidas
- ✓ Descripción del problema
- ✓ **Solución secuencial**
- ✓ Soluciones paralelas y distribuidas
- ✓ Pruebas Realizadas
- ✓ Conclusiones
- ✓ Trabajos Futuros

Solución secuencial

$i-1,j-1$	$i-1,j$	$i-1,j+1$
$i,j-1$	i,j	$i,j+1$
$i+1,j-1$	$i+1,j$	$i+1,j+1$

- ✓ Se representó al océano mediante una matriz de estructuras para mantener el estado de cada posición de la misma.
- ✓ Los pasos que sigue el algoritmo son bastante simples.
 - Inicialización de la matriz
 - Recorrido de la matriz por cada paso de tiempo de la simulación, invocando a una función que analiza y elige aleatoriamente el comportamiento a seguir por el pez o tiburón, dependiendo del estado de las celdas adyacentes y de la acción que se quiere realizar. Efectuando las modificaciones necesarias a las variables de estado.
- ✓ Son verificadas las ocho celdas que rodean a la posición actual, y de todos los movimientos posibles para la acción a realizar, se elige uno aleatoriamente.

Agenda

- ✓ Conceptos de Sistemas Paralelos
- ✓ Conceptos de simulación
- ✓ Simulaciones paralelas y distribuidas
- ✓ Descripción del problema
- ✓ Solución secuencial
- ✓ Soluciones paralelas y distribuidas
- ✓ Pruebas Realizadas
- ✓ Conclusiones
- ✓ Trabajos Futuros

Soluciones paralelas y distribuidas

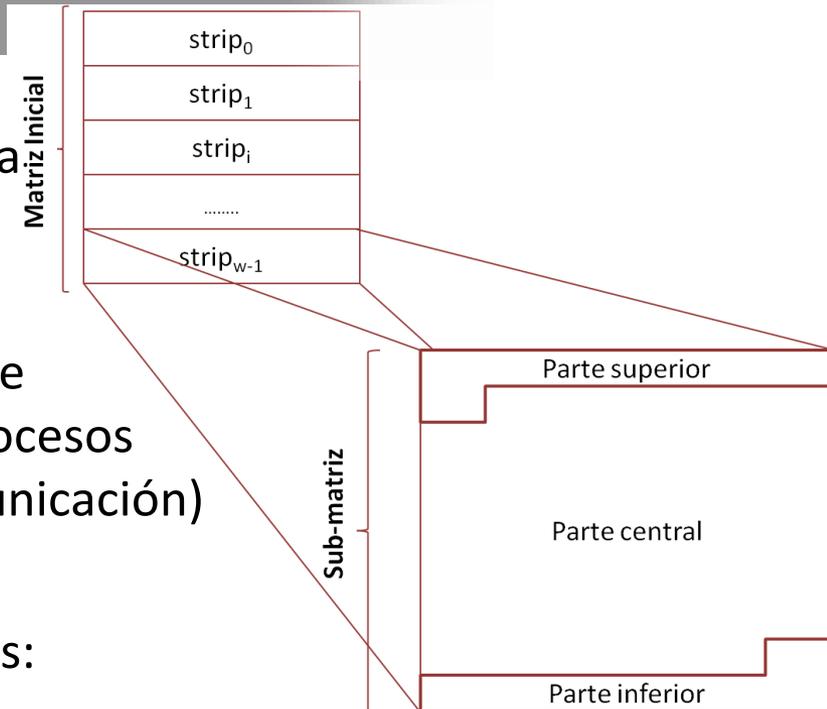
Esquema general de solución

✓ Las soluciones implementadas dividen la matriz en strips horizontales (debido a la homogeneidad).

✓ Como consecuencia de esta división, pueden suceder conflictos en los límites de los fragmentos, ya que se tendrían dos procesos o hilos (dependiendo del modelo de comunicación) trabajando en celdas contiguas.

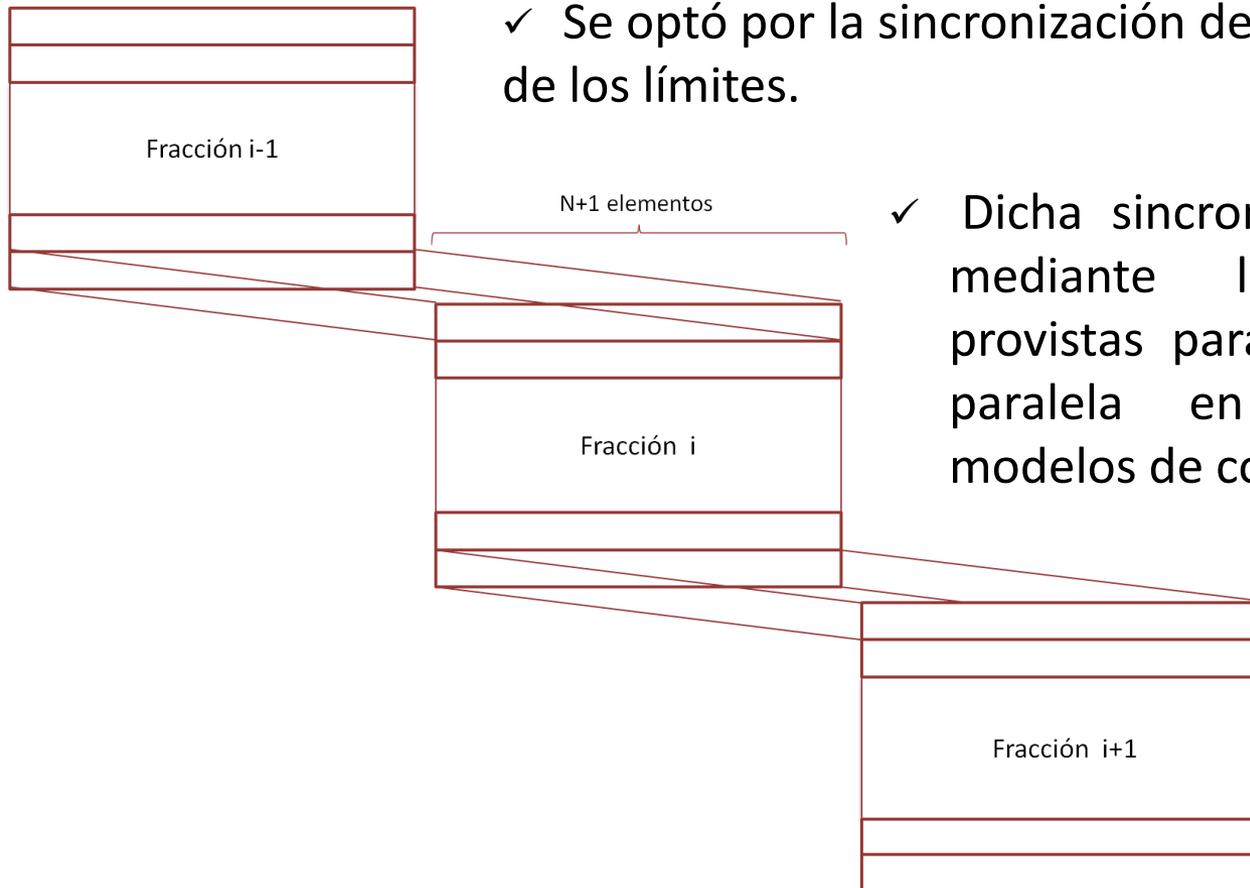
✓ Para su solución existen dos alternativas:

- Un mecanismo que verifique los valores modificados para hacer **rollback** de las operaciones conflictivas.
- Un mecanismo que sincronice la información para evitar los conflictos.



Soluciones paralelas y distribuidas

Esquema general de solución



✓ Se optó por la sincronización del procesamiento de los límites.

✓ Dicha sincronización se realiza mediante las herramientas provistas para la programación paralela en sus diferentes modelos de comunicación.

Soluciones paralelas y distribuidas

Solución de Pasaje de Mensajes

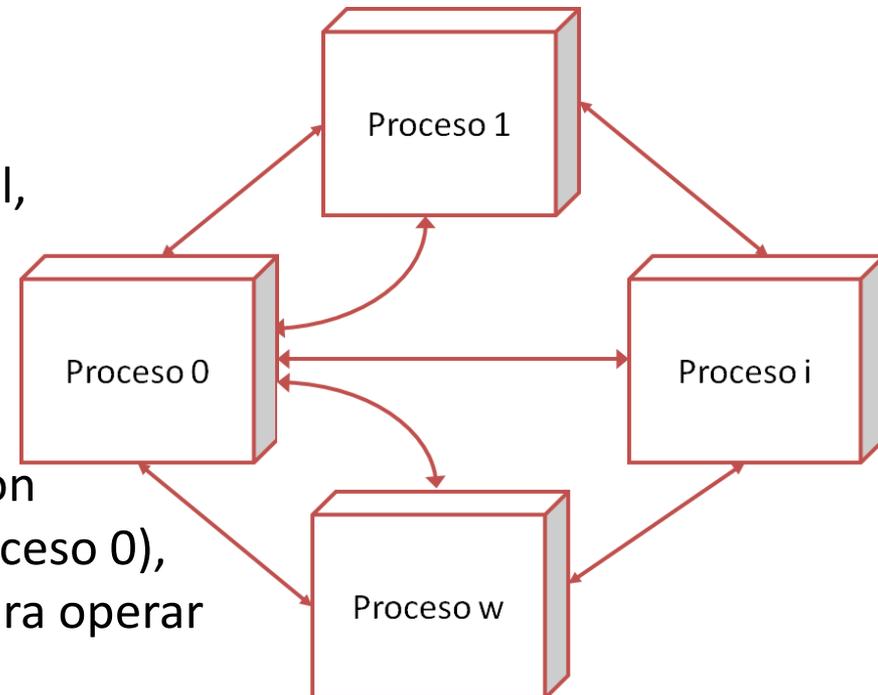
✓ La operatoria del algoritmo se lleva a cabo mediante el uso de una organización circular de procesos.

✓ Cada uno de estos procesos comunica mensajes en forma bidireccional.

✓ Se tienen 2 tipos de procesos:

- El proceso 0 distribuye la matriz inicial, ejecuta la simulación en su fragmento de matriz y recopila el resultado final.

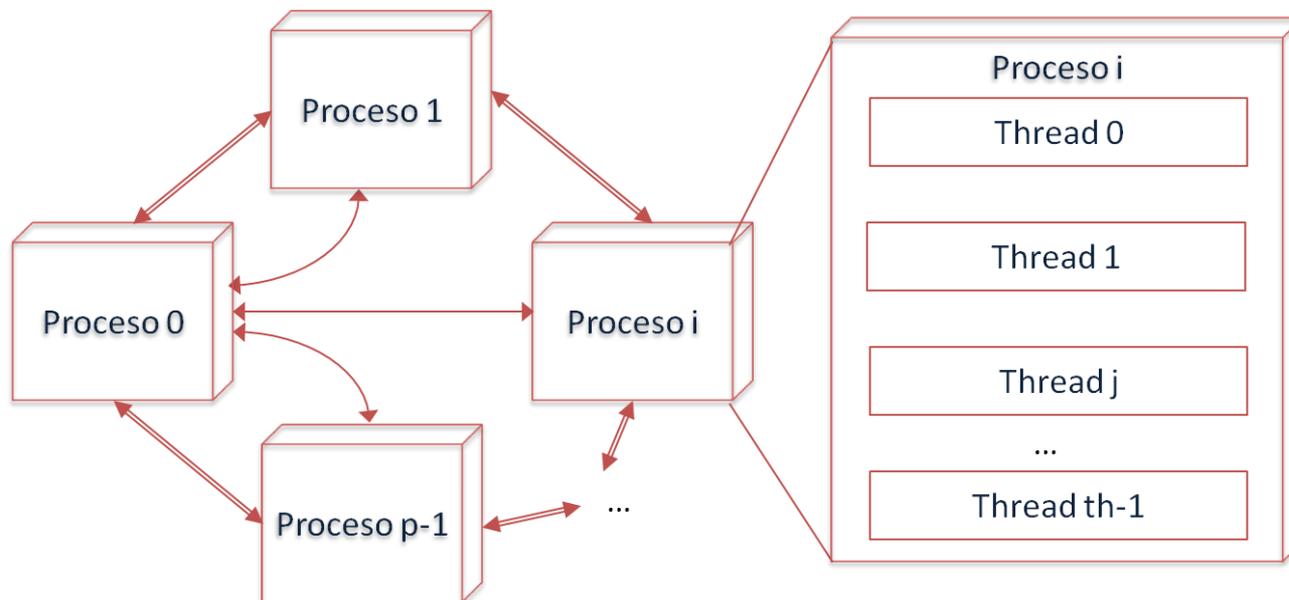
- Los procesos worker reciben su fracción de la matriz y se comunican con los demás workers (también con el proceso 0), enviando los fragmentos necesarios para operar en los límites de las porciones.



Soluciones paralelas y distribuidas

Soluciones híbridas

- ✓ Se realizó un particionamiento de los datos al igual que en el algoritmo MPI, correspondiendo un bloque de filas de la matriz original.
- ✓ Dentro de cada proceso, se repitió el esquema de división de datos de la solución de memoria compartida, correspondiendo a cada hilo una submatriz, sólo que en este caso se le asignará un fragmento correspondiente a la porción del proceso.



Agenda

- ✓ Conceptos de Sistemas Paralelos
- ✓ Conceptos de simulación
- ✓ Simulaciones paralelas y distribuidas
- ✓ Descripción del problema
- ✓ Solución secuencial
- ✓ Soluciones paralelas y distribuidas
- ✓ **Pruebas Realizadas**
- ✓ Conclusiones
- ✓ Trabajos Futuros

Pruebas Realizadas

Descripción de la Plataforma y Escenarios

- ✓ Se utilizó el lenguaje C con las librerías OpenMPI, Pthreads y OpenMP, la primera de ellas para pasaje de mensajes mientras que las dos restantes para memoria compartida.
- ✓ Para realizar la experimentación se ha utilizado una arquitectura Blade de 8 hojas, con 2 procesadores quad core Intel Xeón e5405 de 2.0 GHz en cada una de ellas. Cada hoja posee 2Gb de memoria RAM (compartido entre ambos procesadores) y cache L2 de 2 x 6Mb entre par de núcleos.
- ✓ Para analizar el comportamiento de cada una de las soluciones paralelas implementadas, se mide el tiempo de ejecución, speedup, eficiencia y escalabilidad de las mismas.

Pruebas Realizadas

Descripción de la Plataforma y Escenarios

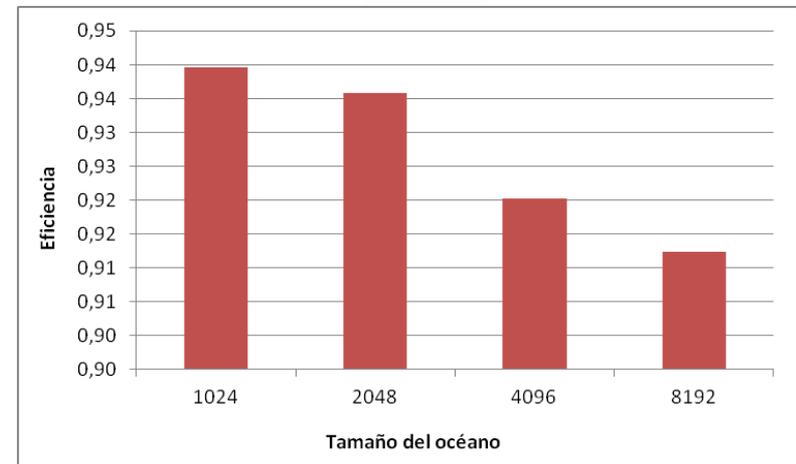
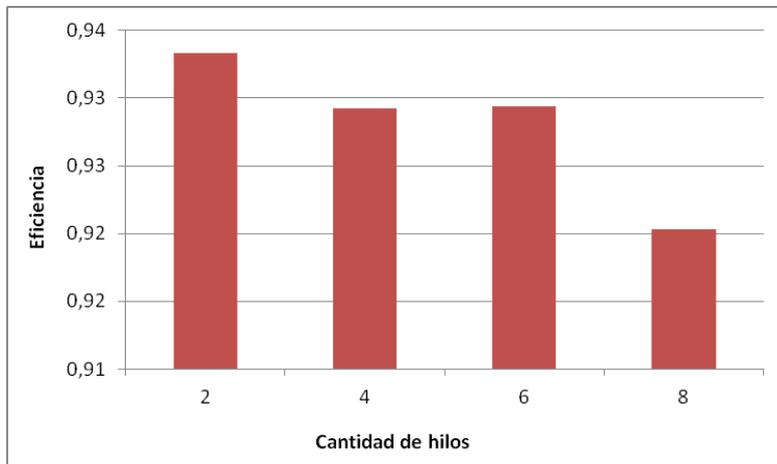
✓ Se obtienen 16 escenarios de prueba variando el tamaño del océano (1024x1024; 2048x2048; 4096x4096; 8192x8192), y modificando el lapso de tiempo de la simulación (1024; 2048; 4096; 8192 momentos).

Escenario	Momento	Tamaño del Océano
1	1024	1024x1024
2	1024	2048x2048
3	1024	4096x4096
4	1024	8192x8192
5	2048	1024x1024
6	2048	2048x2048
7	2048	4096x4096
8	2048	8192x8192

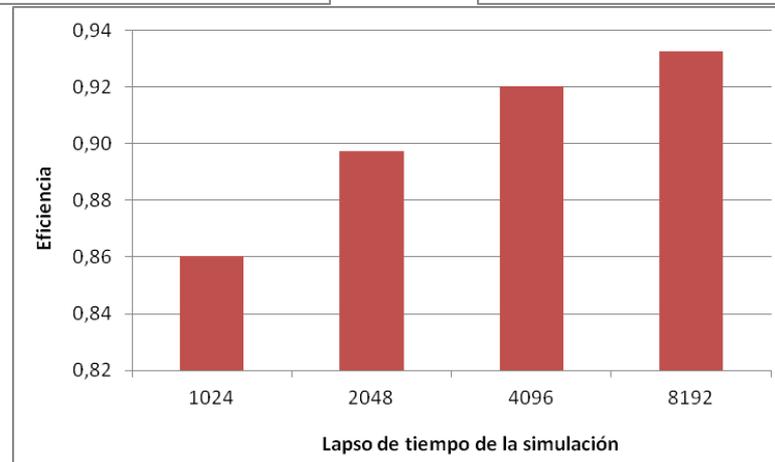
Escenario	Momento	Tamaño del Océano
9	4096	1024x1024
10	4096	2048x2048
11	4096	4096x4096
12	4096	8192x8192
13	8192	1024x1024
14	8192	2048x2048
15	8192	4096x4096
16	8192	8192x8192

Pruebas Realizadas

Resultados de Memoria Compartida - Pthreads

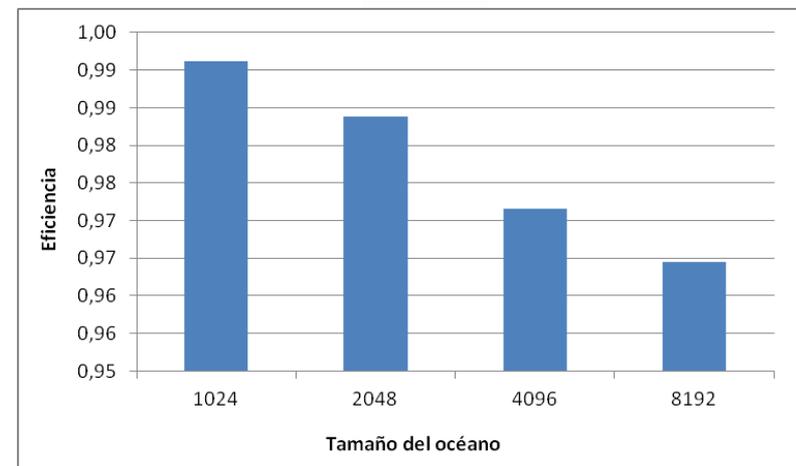
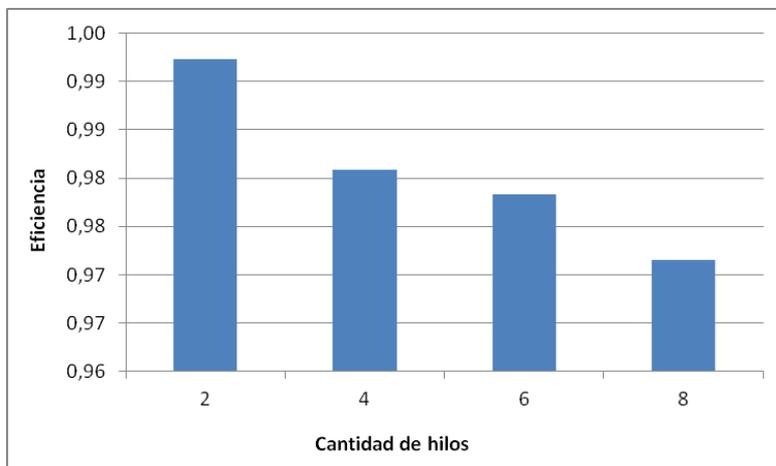


Matriz de 4096x4096
4096 generaciones
8 hilos

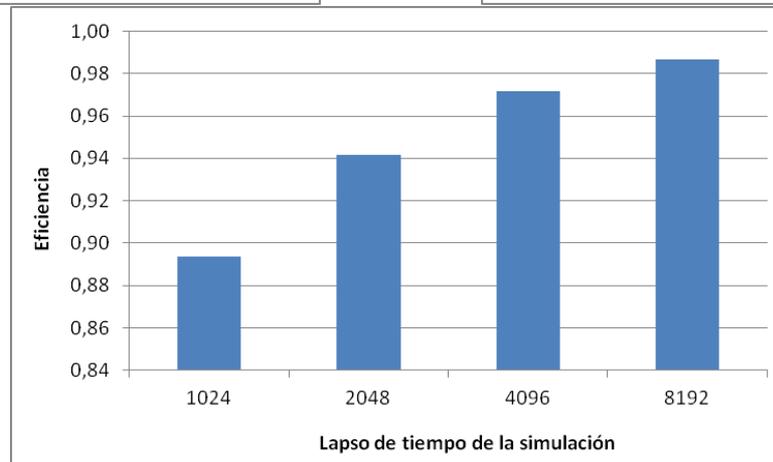


Pruebas Realizadas

Resultados de Memoria Compartida - OpenMP

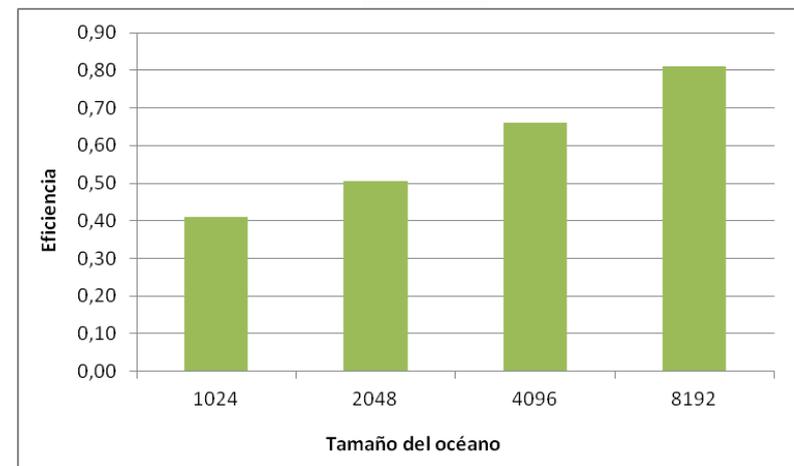
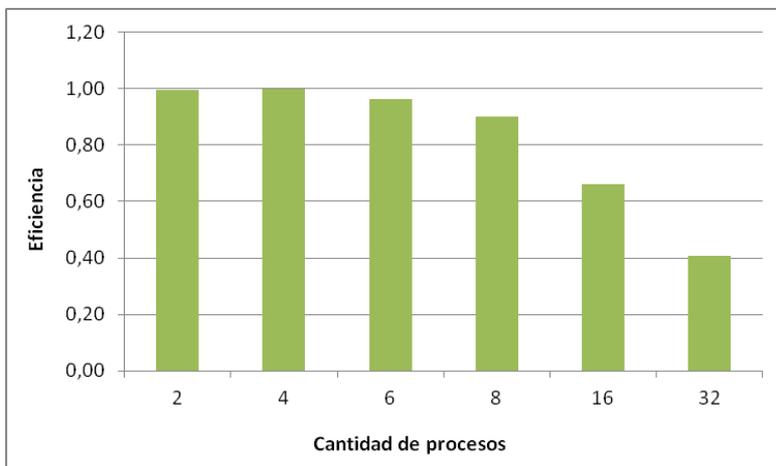


Matriz de 4096x4096
4096 generaciones
8 hilos

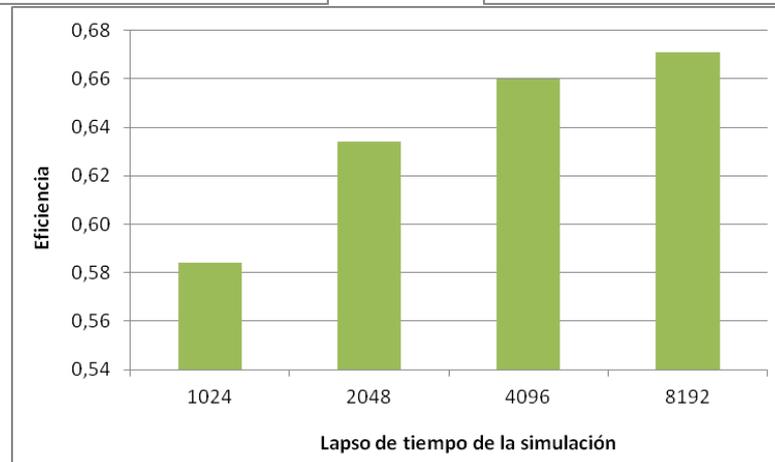


Pruebas Realizadas

Resultados de Pasaje de Mensajes

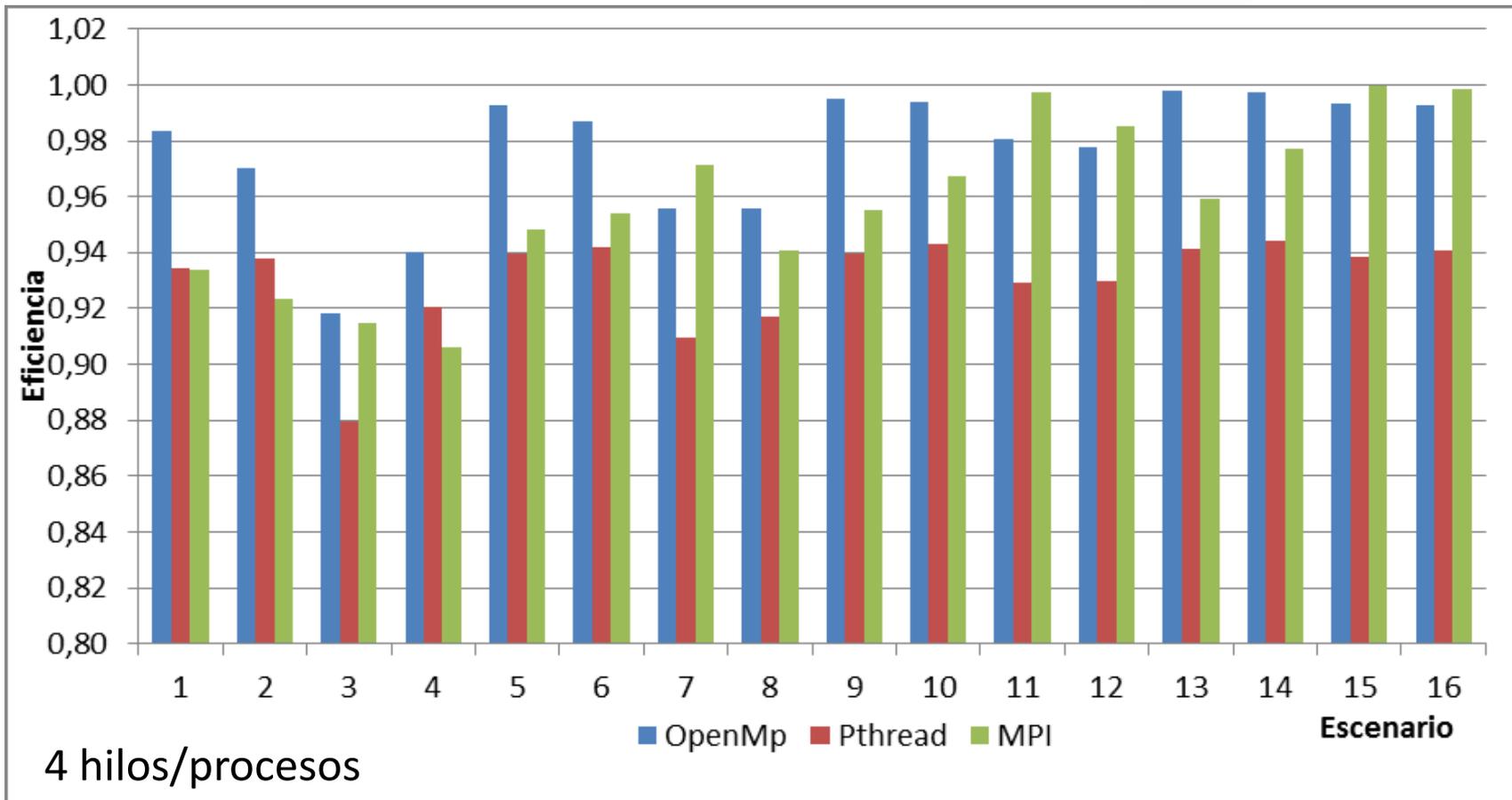


Matriz de 4096x4096
4096 generaciones
16 procesos



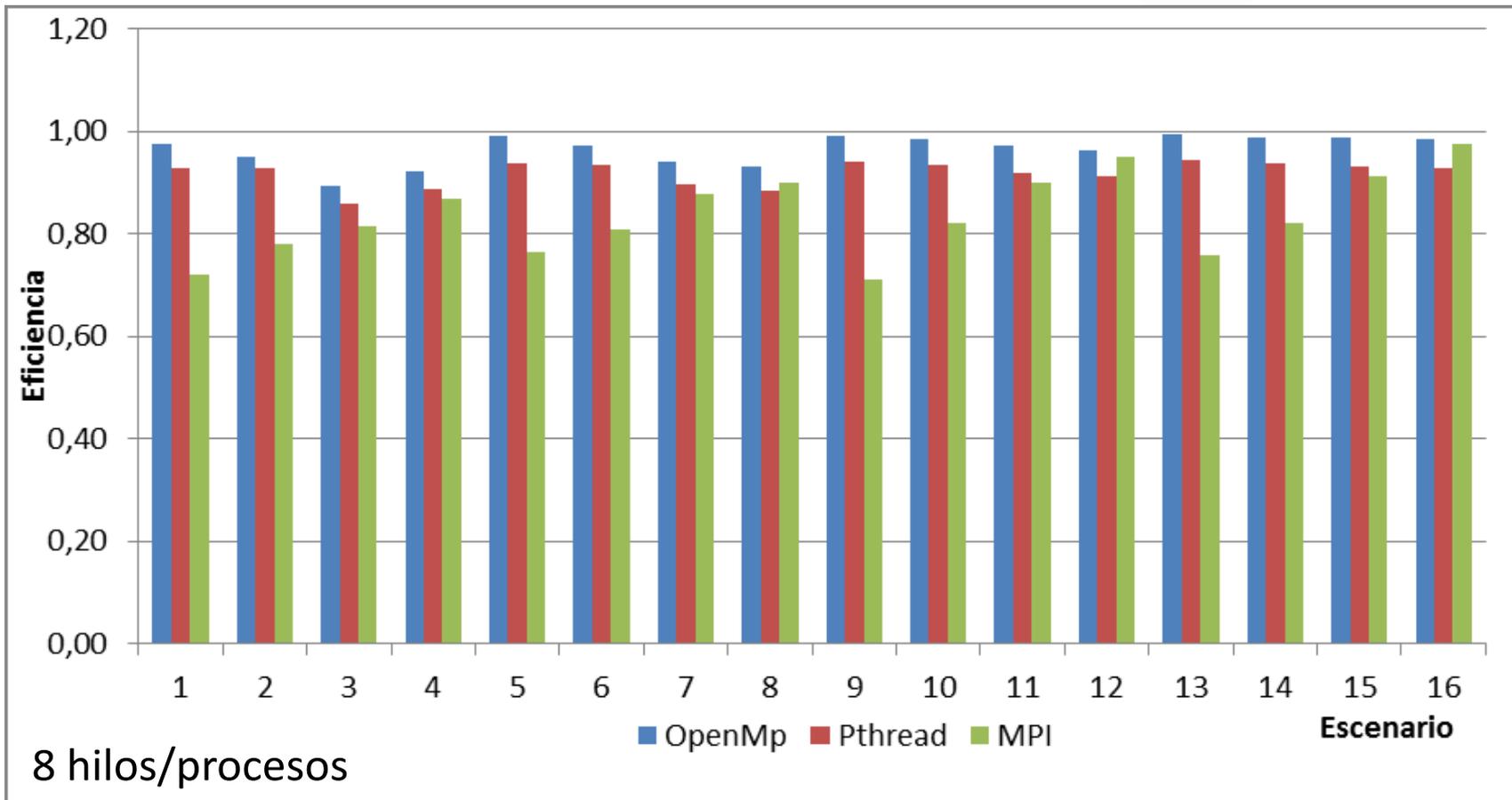
Pruebas Realizadas

Comparación de las Soluciones Puras



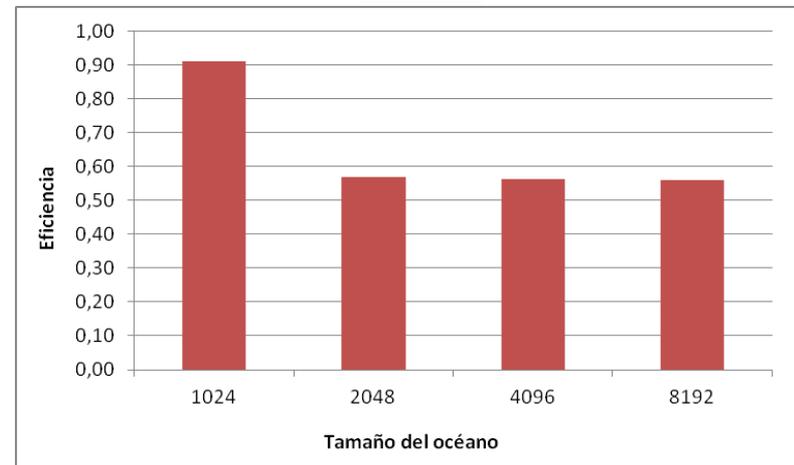
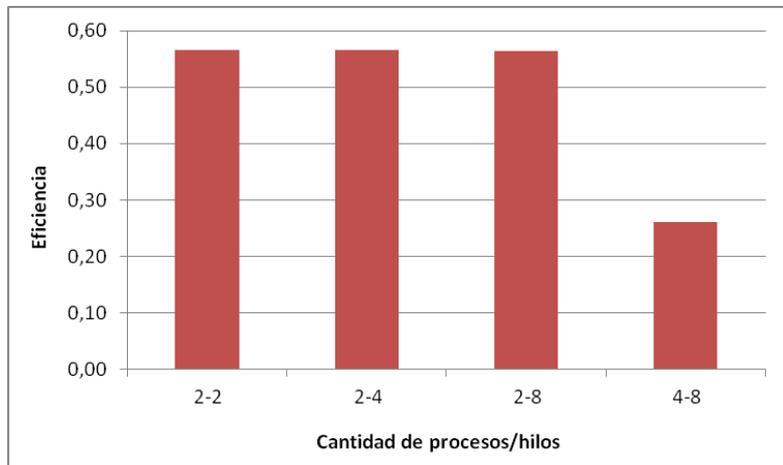
Pruebas Realizadas

Comparación de las Soluciones Puras

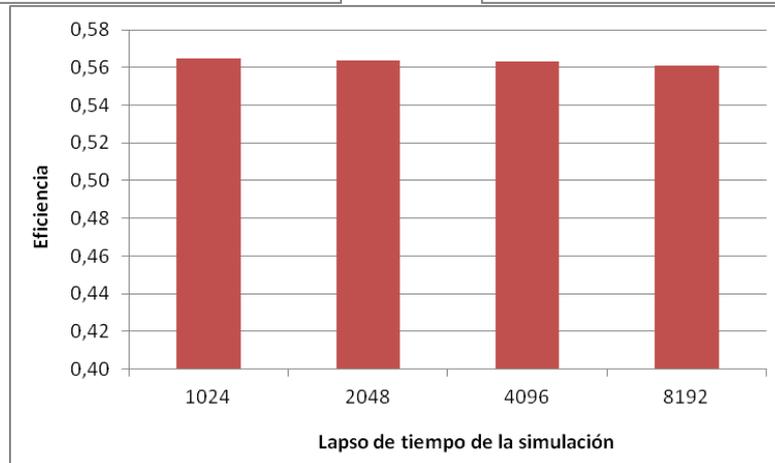


Pruebas Realizadas

Resultados de las Soluciones Híbridas (MPI-Pthreads)

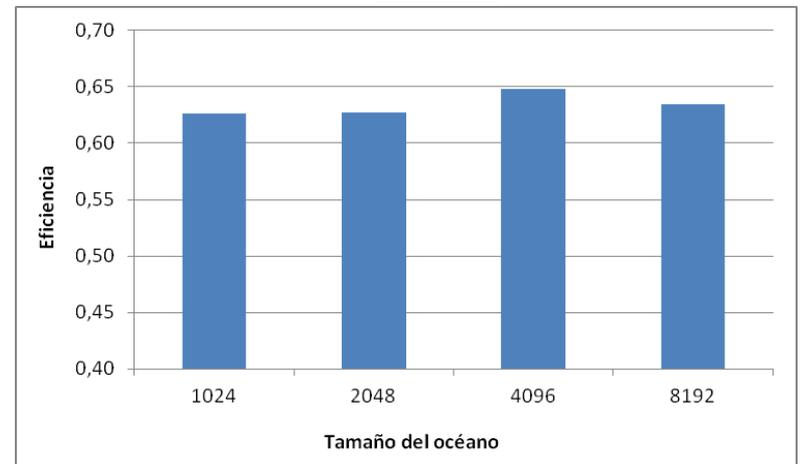
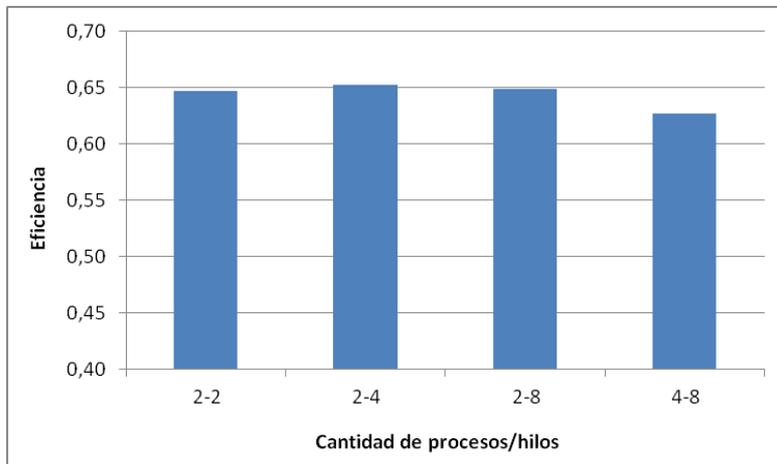


Matriz de 4096x4096
4096 generaciones
2-8 procesos/hilos

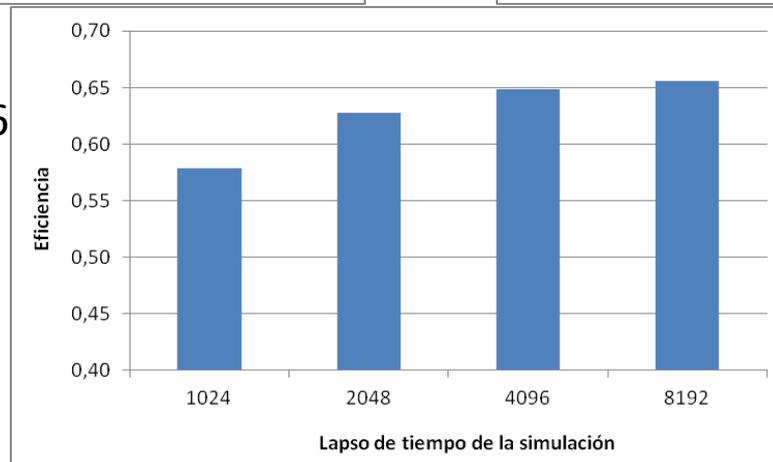


Pruebas Realizadas

Resultados de las Soluciones Híbridas (MPI-OpenMP)

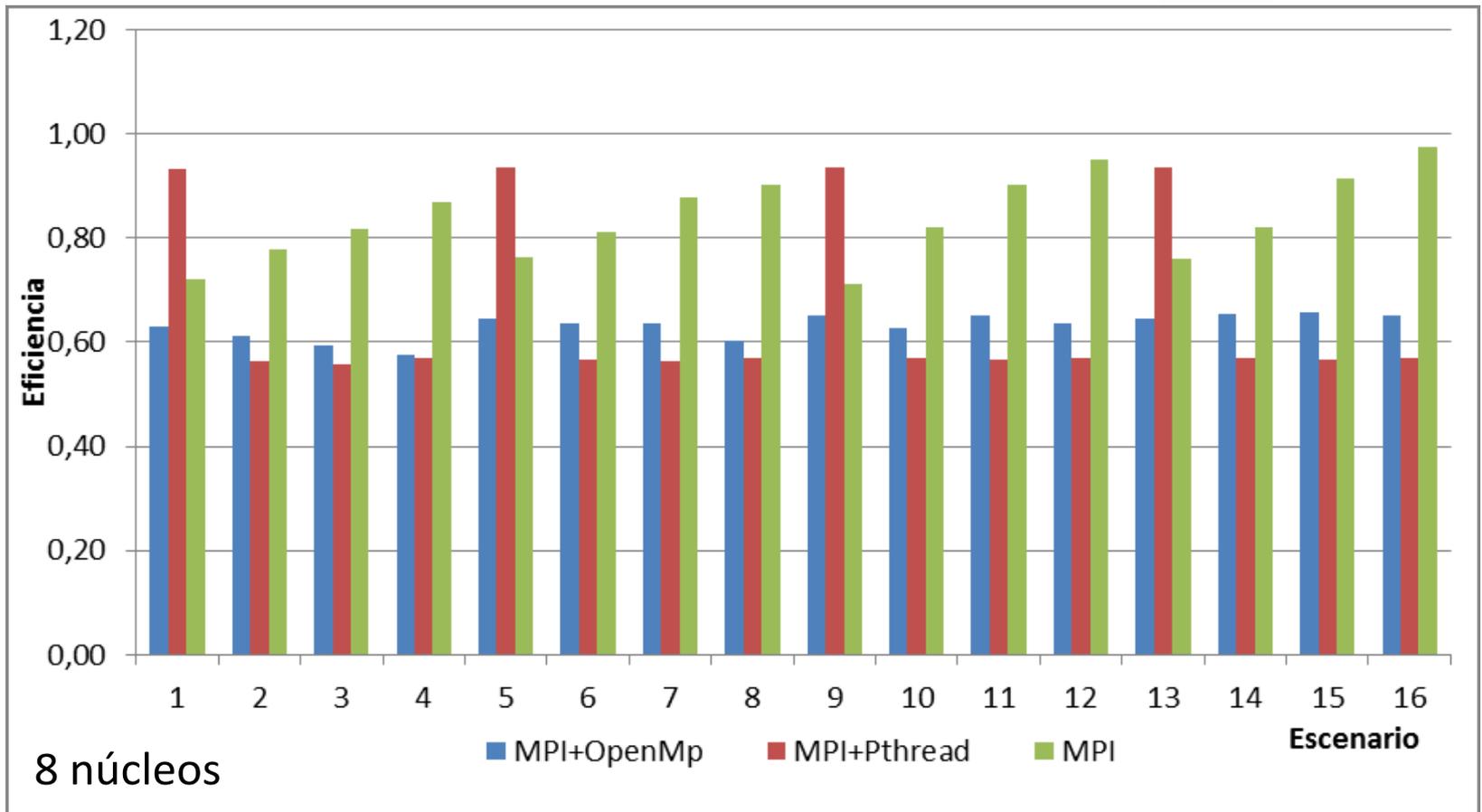


Matriz de 4096x4096
4096 generaciones
2-8 procesos/hilos



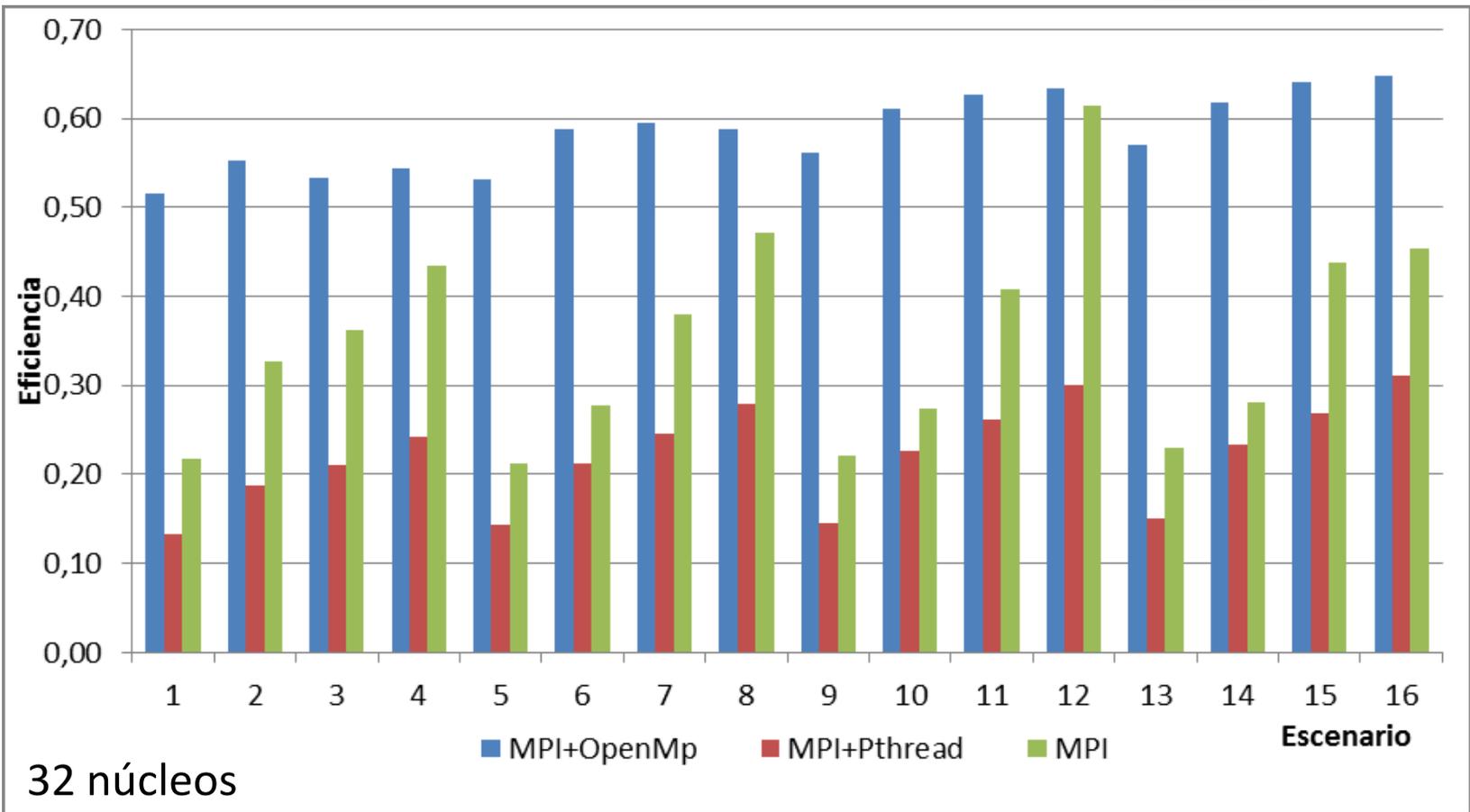
Pruebas Realizadas

Comparación entre Soluciones Híbridas y PM



Pruebas Realizadas

Comparación entre Soluciones Híbridas y PM



Agenda

- ✓ Conceptos de Sistemas Paralelos
- ✓ Conceptos de simulación
- ✓ Simulaciones paralelas y distribuidas
- ✓ Descripción del problema
- ✓ Solución secuencial
- ✓ Soluciones paralelas y distribuidas
- ✓ Pruebas Realizadas
- ✓ **Conclusiones**
- ✓ Trabajos Futuros

Conclusiones

- ✓ Se realizó Investigación y Desarrollo de algoritmos paralelos sobre arquitecturas de soporte actuales de múltiples núcleos, con aplicación a problemas de simulaciones con alta demanda computacional.
- ✓ Se plantearon diferentes soluciones paralelas utilizando distintos modelos de comunicación:

Soluciones Puras			Soluciones híbridas	
Memoria Compartida		Pasaje de Mensajes		
OpenMP	Pthreads	OpenMPI	OpenMPI-OpenMP	OpenMPI-Pthreads

Conclusiones

- ✓ En cuanto a las soluciones con memoria compartida, la implementación OpenMP se comportó con mejores tiempos de ejecución que la solución Pthreads.
- ✓ Si se comparan las soluciones “puras”, cuando la cantidad de hilos/procesos no es grande (2 y 4), en general la eficiencia lograda con OpenMP y MPI es similar y superior a la de Pthreads.
- ✓ Además cuanto más grande es el problema (tamaño del océano y lapso de tiempo de la simulación) más se nota la aproximación de los resultados de MPI a los de OpenMP, y en algunos casos incluso la superioridad.
- ✓ Las soluciones de memoria compartida se comportan (en general) mejor que las de Pasaje de Mensajes cuando se trabaja con una mayor cantidad de hilos/procesos (6 y 8). Esto se debe en parte a que se puede aprovechar eficientemente los niveles de memoria compartida de la arquitectura teniendo en cuenta que este tipo de problema requiere extensivo uso de memoria, evitando de esta manera las comunicaciones explícitas y sincronizaciones implícitas del PM.

Conclusiones

- ✓ En cuanto a las soluciones híbridas, como era esperable la que utiliza OpenMPI + OpenMP se comporta de manera más eficiente que la de OpenMPI + Pthreads.
- ✓ Si se comparan las soluciones híbridas con las de Pasaje de Mensajes, a medida que se utiliza una arquitectura con más núcleos, en la solución híbrida se observa que si bien en ambos casos la eficiencia cae, este decremento es mayor en el caso de Pasaje de Mensajes. De esta forma, la solución híbrida se comporta mejor.



Agenda

- ✓ Conceptos de Sistemas Paralelos
- ✓ Conceptos de simulación
- ✓ Simulaciones paralelas y distribuidas
- ✓ Descripción del problema
- ✓ Solución secuencial
- ✓ Soluciones paralelas y distribuidas
- ✓ Pruebas Realizadas
- ✓ Conclusiones
- ✓ Trabajos Futuros

Trabajos Futuros

- ✓ Extender el estudio de escalabilidad para tamaños de problema y arquitecturas más grandes.
- ✓ Estudiar alternativas de solución mediante estrategias optimistas.
- ✓ Extender la aplicación de las estrategias y los resultados obtenidos a otros problemas de cómputo científico que presenten características similares.
- ✓ Incorporar heterogeneidad en la arquitectura, estudiando el impacto sobre el mapeo de procesos y datos a procesadores.

