

Diseño de Software Paralelo

Jorge L. Ortega Arjona

Departamento de Matemáticas

Facultad de Ciencias, UNAM

jloa@fciencias.unam.mx

**Ingeniería de
Sistemas y
Redes**

Programación...

Arquitectura de...

Parallel Pipes...

Communicating...

Manager-...

Patterns for...

Referencias

Home Page

Título



Atrás

Página 1 de 22

Pant. Completa

Cerrar

Salir

LA TEXiendo
100cia

1. Programación Paralela

Programa Paralelo: *“Especificación de un conjunto de procesos que se ejecutan simultáneamente, comunicándose entre sí para lograr un objetivo común”.*

- *Ventaja:* Habilidad para manejar tareas de una escala que no sería realista o costo-efectivo para otros sistemas de cómputo.
- *Desventaja:* Diseñar y programar software paralelo es, en general, una labor difícil.

El objetivo principal de cualquier Programa Paralelo para la mayoría de científicos e ingenieros es mejorar su desempeño, es decir, el tiempo de ejecución de una aplicación.

Programación . . .

Arquitectura de . . .

Parallel Pipes . . .

Communicating . . .

Manager- . . .

Patterns for . . .

Referencias

Home Page

Título



Atrás

Página 2 de 22

Pant. Completa

Cerrar

Salir

1.1. Problemas a resolver con Paralelismo

- ¿Para qué se usa la Programación Paralela?
- ¿Hay problemas que necesiten algo más que la Programación Paralela?
- Algunas áreas de aplicación son:
 - Predicción del clima.
 - Cosmología
 - Dinámica de fluidos.
 - Ciencia de materiales.
 - Química cuántica.
 - Biología y bioquímica.
 - Cómputo Visual y realidad artificial.
 - Inteligencia artificial y vida artificial.

Programación . . .

Arquitectura de . . .

Parallel Pipes . . .

Communicating . . .

Manager- . . .

Patterns for . . .

Referencias

Home Page

Título



Atrás

Página 3 de 22

Pant. Completa

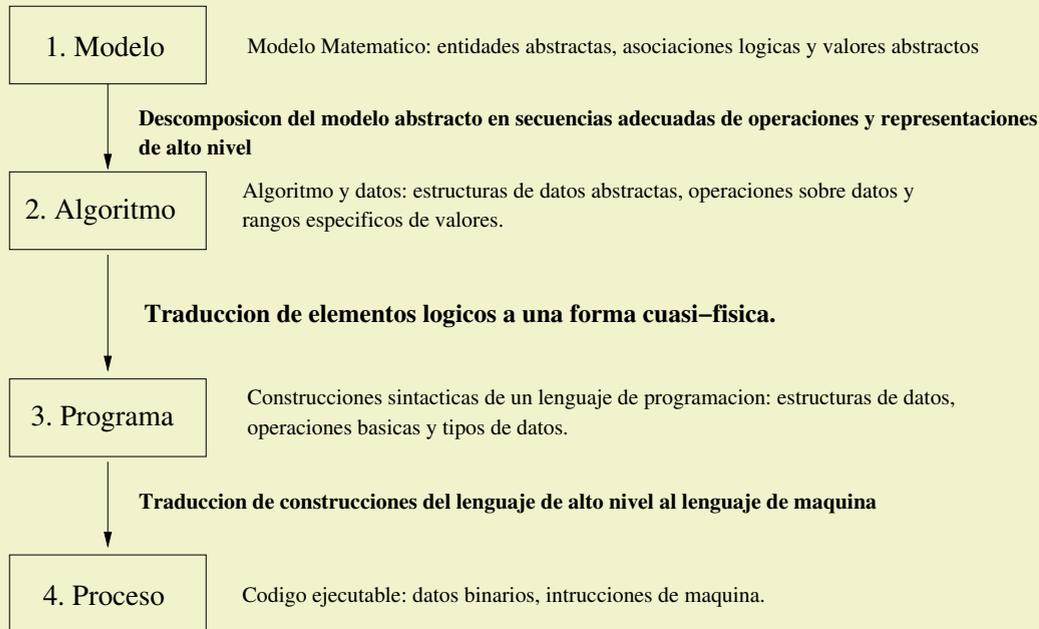
Cerrar

Salir

1.2. Factores que influyen el desempeño de un programa paralelo [Pancake, 1996]

- Plataforma de Hardware
 - Número y calidad de procesadores
 - Cantidad de memoria de trabajo disponible
 - Número y calidad de componentes de comunicación
 - etc.
- Lenguaje de Programación
 - Expresión del paralelismo
 - Expresión de la secuencialidad
 - Instrucciones de Comunicación
 - Manejo del No-determinismo
- Problema a Resolver

1.3. El Problema a Resolver [Pancake & Bergmark, 1990]



- ¿Cómo particionar algoritmo?, ¿Cómo particionar los datos?
- ¿Qué estructura de programación considerar, dada la plataforma de hardware, el lenguaje de programación y estos particionamientos?

Programación . . .

Arquitectura de . . .

Parallel Pipes . . .

Communicating . . .

Manager- . . .

Patterns for . . .

Referencias

Home Page

Título



Atrás

Página 5 de 22

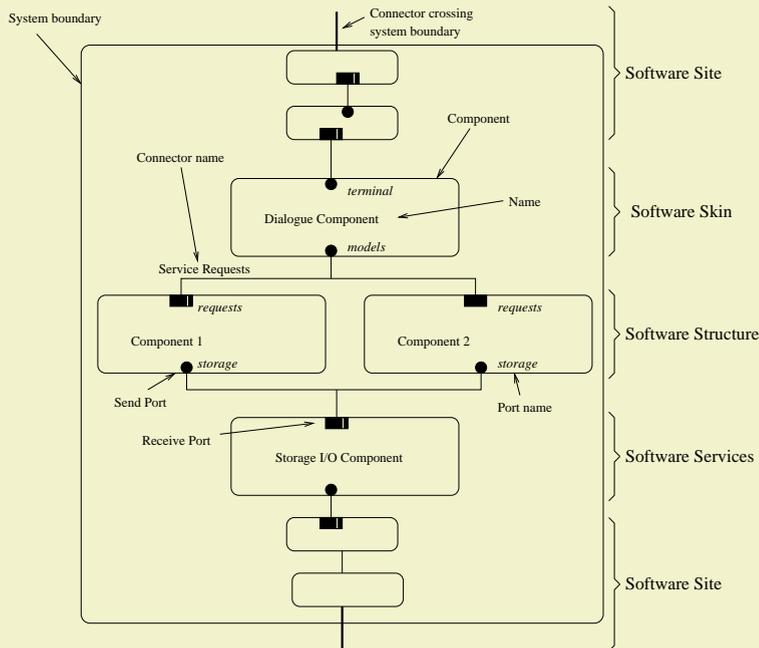
Pant. Completa

Cerrar

Salir

2. Arquitectura de Software

Arquitectura de Software: *“Una arquitectura de software es una descripción de los sub-sistemas y componentes de un sistema de software, y las relaciones entre ellos”* [Buschmann, et al. 1996]



2.1. Descripción de la Arquitectura de Software usando Patrones Arquitectónicos

Patrón Arquitectónico: *“Un patrón arquitectónico expresa un esquema de organización estructural fundamental para sistemas de software. Provee de un conjunto de sub-sistemas predefinidos, especifica sus responsabilidades, e incluye reglas y guías para organizar la relación entre ellos”* [Buschmann, et al. 1996]

- **Estructura:** Forma en que se compone el sistema de software.
 - Componentes: unidades básicas de software que realizan el procesamiento.
 - Conexiones: unidades básicas de software que permiten el intercambio de información.
- **Dinámica:** Relación (comunicación) entre los componentes que lleva a cabo la función del sistema de software.

Home Page

Título



Atrás

Página 7 de 22

Pant. Completa

Cerrar

Salir

2.2. Patrones Arquitectónicos para Programación Paralela [Ortega & Roberts, 1998]

- Paralelismo funcional (partición del algoritmo):
 - *Parallel Pipes and Filters*
 - *Parallel Hierarchies*
- Paralelismo de dominio (partición de los datos):
 - *Communicating Sequential Elements*
- Paralelismo de actividad (partición del algoritmo y los datos):
 - *Manager-Workers*
 - *Shared Resource*

Home Page

Título



Atrás

Página 8 de 22

Pant. Completa

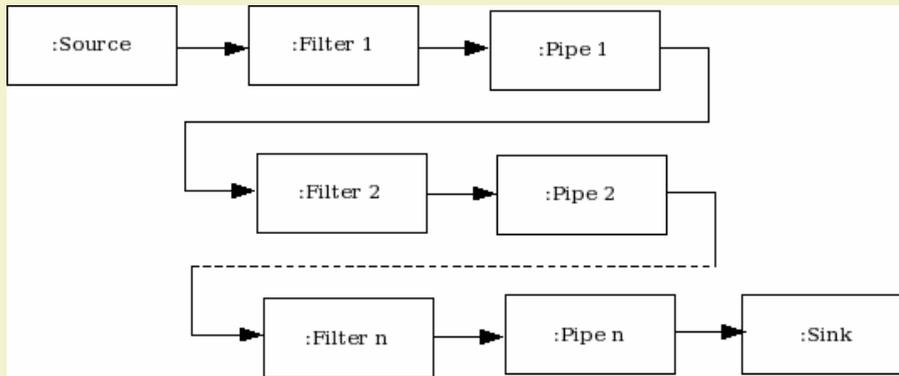
Cerrar

Salir

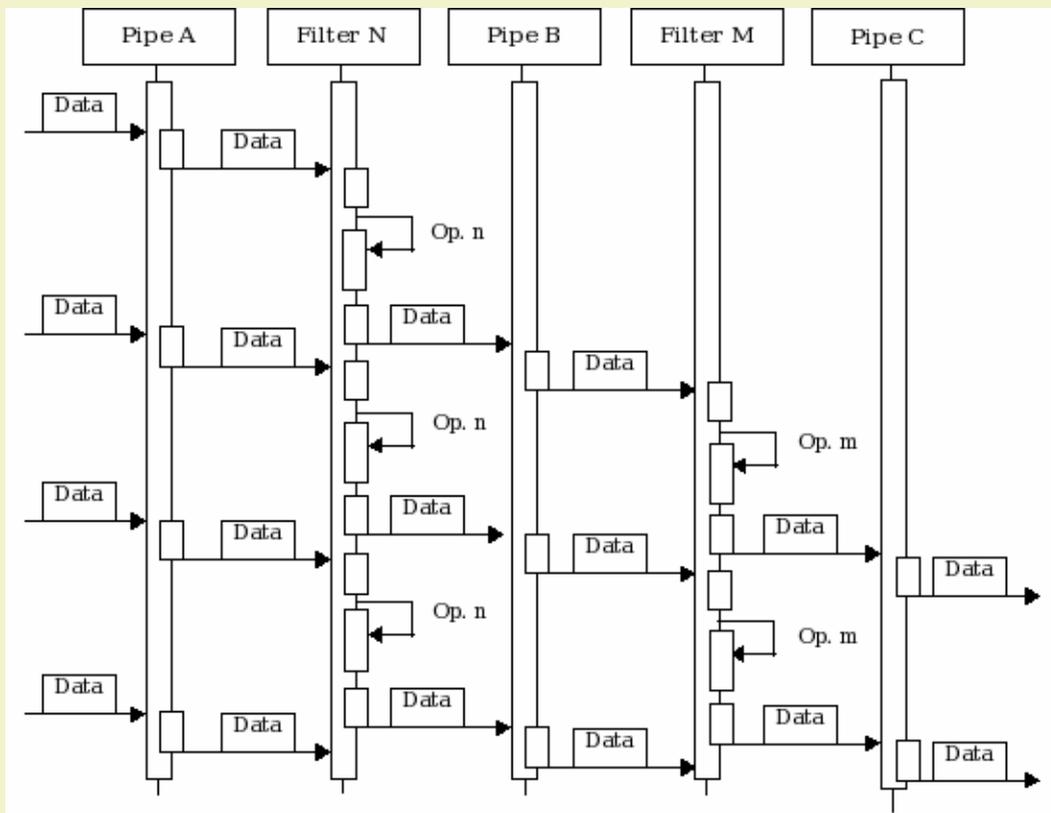
3. Parallel Pipes and Filters [Ortega, 2005]

El patrón *Parallel Pipes and Filters* extiende al patrón *Pipes and Filters* original [Buschmann, et al., 1996] con aspectos de paralelismo funcional. Cada componente paralelo simultáneamente realiza un paso diferente del cómputo, siguiendo un orden preciso de operaciones sobre datos ordenados que se pasan de un paso a otro, en forma de un flujo de datos a través de toda la estructura.

3.1. Estructura



3.2. Dinámica



3.3. Aplicación en Visualización: 3D Rendering

Problema:

- *3D Rendering* se refiere al conjunto de operaciones necesarias para proyectar la vista de un objeto o escena tridimensional sobre una superficie.
- Las etapas comunes de un proceso de *rendering* son:
 1. enumerar los objetos en la escena y generar descripciones geométricas de cada uno;
 2. aplicar una transformación a la geometría para tomar en cuenta la posición de la cámara, la dirección y longitud focal;
 3. hacer cortes a la geometría para que los elementos fuera del campo de visión se excluyan;
 4. aplicar una iluminación para obtener las sombras de cada objeto; y
 5. rastrear y convertir las caras de los objetos, a fin de obtener una imagen final.
- En aplicaciones comunes en la industria fílmica, realizar el *rendering* de la escena de un efecto especial de 10 segundos de duración usando una resolución estándar de 2048x1536 pixeles toma alrededor de 130 horas tiempo de procesamiento en una sola plataforma de alto rendimiento Macintosh o PC.

Home Page

Título



Atrás

Página 11 de 22

Pant. Completa

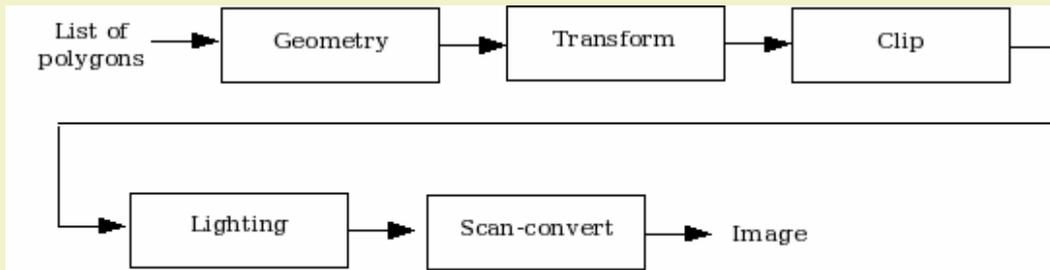
Cerrar

Salir



Solución:

Se puede realizar este procesamiento más eficientemente mediante traslapar cada paso del rendering en el tiempo, en forma de una arquitectura *Pipes and Filters*:



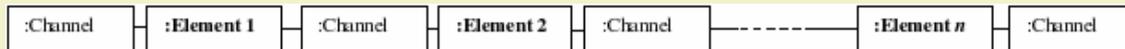
Usando una aproximación paralela, con una plataforma CYCORE (una máquina paralela de Parsytec) de 16 procesadores, el tiempo de procesamiento de la escena se reduce a 10.5 horas.

4. Communicating Sequential Elements

[Ortega, 2000]

Communicating Sequential Elements es un patrón de paralelismo de dominio en el cual cada componente realiza la misma operación sobre diferentes trozos de una estructura regular de datos. La operación en cada componente depende de los resultados parciales de sus componentes vecinos. Normalmente, este patrón se concibe como una estructura lógica, reflejando el orden presente en los datos del problema.

4.1. Estructura



Home Page

Título



Atrás

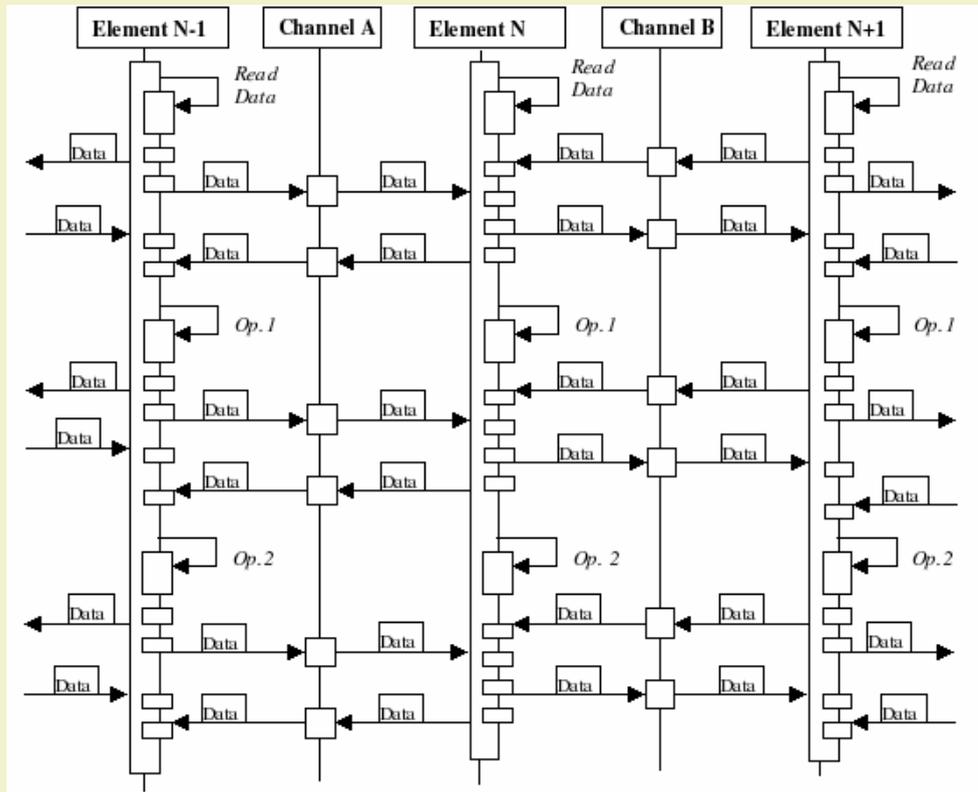
Página 13 de 22

Pant. Completa

Cerrar

Salir

4.2. Dinámica



4.3. Aplicación en Modelación de Materiales: La Ecuación de Calor

Problema:

- Calor es el nivel de energía de un cuerpo físico, perceptible por su temperatura.
- La temperatura en una parte del cuerpo varía a través del tiempo, incrementándose o decrementándose por intercambio de calor entre las partes.
- En un caso unidimensional (un alambre, por ejemplo), la Ecuación de Calor tiene la siguiente forma discreta:

$$A[i + 1, j] = A[i, j] + \frac{\Delta t}{\Delta x^2} (A[i, j + 1] - 2A[i, j] + A[i, j - 1])$$

La solución numérica calcula el valor de cada segmento (j) a un tiempo dado (i) considerando su temperatura previa las temperaturas de los segmentos vecinos.

- El tiempo total para obtener la solución numérica depende (a) del número de segmentos y (b) del número de pasos de tiempo requeridos.

Home Page

Título



Atrás

Página 15 de 22

Pant. Completa

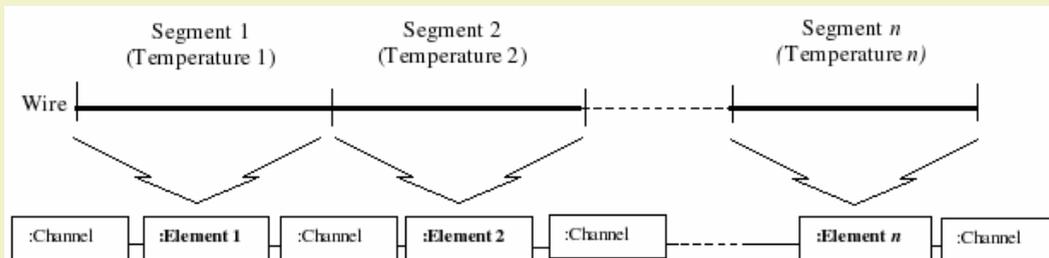
Cerrar

Salir

Solución:

Se puede ejecutar todo este cómputo más eficientemente mediante:

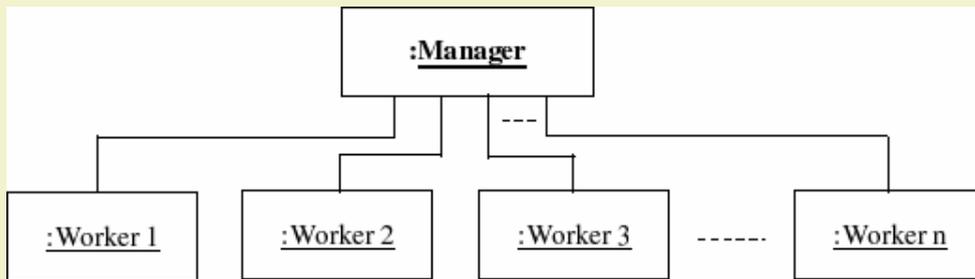
- usar un conjunto de componentes en paralelo que exploten la estructura lógica unidimensional del alambre, y
- calcular simultáneamente en un paso el valor de $A[i + 1, j]$ para todos los segmentos.



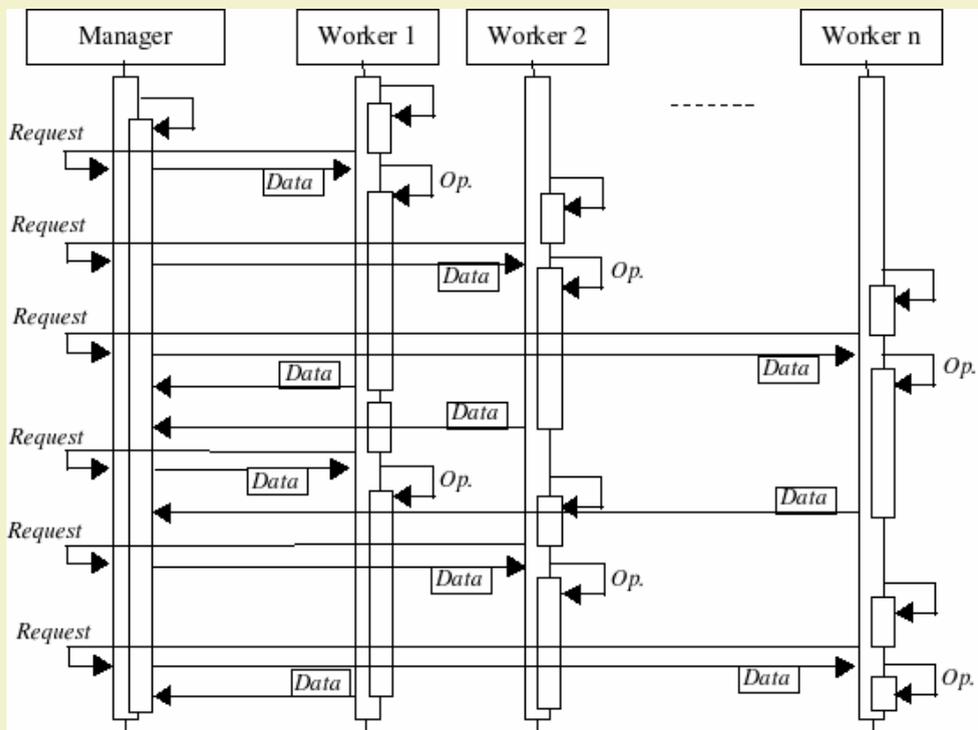
5. Manager-Workers [Ortega, 2004]

El patrón *Manager-Workers* es una variante del patrón *Master-Slave* [Buschmann, et al. 1996] para sistemas paralelos, considerando una aproximación de paralelismo de actividad donde la misma operación se realizan sobre datos ordenados. La variación se basa en el hecho de que los componentes de este patrón son proactivos en lugar de reactivos: cada componente realiza simultáneamente la misma operación, independientemente de la actividad de procesamiento de los otros componentes. Una característica importante es que el orden de los datos debe preservarse en el resultado final.

5.1. Estructura



5.2. Dinámica

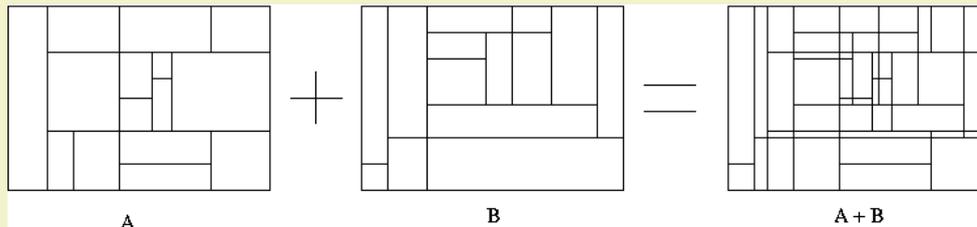




5.3. Aplicación en Visualización: El Problema de Superposición de Polígonos

Problema:

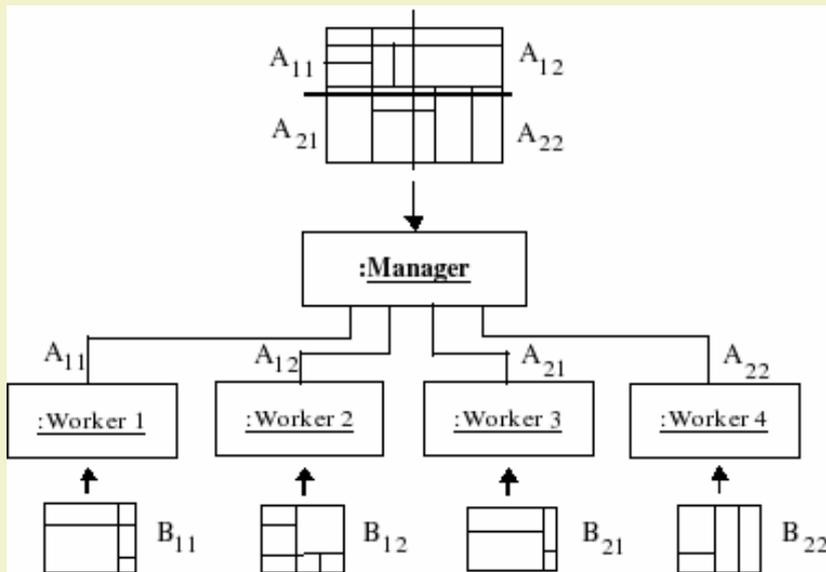
- El Problema de Superposición de Polígonos (*Polygon Overlay Problem*) consiste en obtener la superposición de dos mapas rectangulares **A** y **B**, de la misma área, y compuesto por polígonos rectangulares no superpuestos.
- Este problema es común en sistemas de información geográfica, donde es necesario visualizar en un solo mapa, por ejemplo, tipo de suelo y vegetación. La superposición muestra la distribución de la combinación tipo de suelo y vegetación.



- La solución secuencial toma un polígono de **A**, y busca todas las superposiciones con todos los polígonos de **B**. Esto es efectivo, pero lento para grandes números de polígonos.

Solución:

Ya que la operación de superposición entre dos polígonos es independiente de la superposición de otros dos polígonos, se puede realizar simultáneamente muchas superposiciones.



6. Patterns for Parallel Software Design



Marzo, 2010.

Programación...

Arquitectura de...

Parallel Pipes...

Communicating...

Manager...

Patterns for...

Referencias

Home Page

Título



Atrás

Página 21 de 22

Pant. Completa

Cerrar

Salir

7. Referencias

- Buschmann, F., Meunier, R., Rohnert, H., Sommerland, P., and Stal, M. (1996) *Pattern-Oriented Software Architecture. A System of Patterns*. John Wiley & Sons, Ltd. Chichester, United Kingdom.
- Ortega-Arjona, J.L., and Roberts, G.R. (1998) *Architectural Patterns for Parallel Programming*. Proceedings of the 3rd European Conference On Pattern Languages of Programming and Computing, EuroPLoP'98.
- Ortega-Arjona, J.L. (2000) *The Communicating Sequential Elements Pattern. An Architectural Pattern for Domain Parallelism*. Proceedings of the 7th Conference on Pattern Languages of Programming, PLoP 2000.
- Ortega-Arjona, J.L. (2004) *The Manager-Workers Pattern. An Activity Parallelism Architectural Pattern for Parallel Programming*. Proceedings of the 9th European Conference on Pattern Languages of Programming and Computing, EuroPLoP 2004.
- Ortega-Arjona, J.L. (2005) *The Pipes and Filters Pattern. A Functional Parallelism Architectural Pattern for Parallel Programming*. Proceedings of the 10th European Conference on Pattern Languages of Programming and Computing, EuroPLoP 2005.
- Pancake, C. M. (1996) *Is Parallelism for You?* Oregon State University. Originally published in Computational Science and Engineering, Vol. 3, No. 2.
- Pancake, C.M., and Bergmark, D. (1990) *Do Parallel Languages Respond to the Needs of Scientific Programmers?* IEEE Computer 23(12).

Home Page

Título



Atrás

Página 22 de 22

Pant. Completa

Cerrar

Salir