

# **PROGRAMACION VECTORIAL**

**RAFAEL VALDÉS VALDAZO UO196581  
ÁNGEL MARÍA VILABOA PÉREZ UO197092  
BLOQUE PARALELAS 4º INFORMÁTICA  
UNIVERSIDAD DE OVIEDO**

# INTRODUCCIÓN

- Ligada al concepto de vector
  - Decodifica instrucciones cuyos operandos sean vectores.
  - Realiza operaciones aritmético-lógicas sobre las componentes de dichos vectores.
- Nuevo juego de instrucciones
- Ejemplos:
  - Nemotécnico VADD-> suma de vectores.
  - Nemotécnico VSUM-> suma de componentes



# CARACTERÍSTICAS DEL LENGUAJE

- Deben poseer un alto grado de flexibilidad para declarar diferentes clases de objetos con distintas estructuras y formas de almacenamiento.
- El lenguaje debe ser eficaz para la manipulación de matrices y vectores dispersos.
- Deben disponer de operaciones vectoriales nativas que trabajen directamente sobre las estructuras de datos comentadas anteriormente sin necesidad de bucles.



# VENTAJAS

- Elimina conflictos debido a dependencias de datos.
- Nos reduce el efecto del cuello de botella.
- Acceso a memoria en las instrucciones vectoriales mediante un patrón fijo lo que facilita su lectura.
- Se reducen los riesgos de control que podrían surgir en los saltos de bucle.



# CONVERSIÓN DE CÓDIGO ESCALAR EN CÓDIGO VECTORIAL

- Dos formas de generar código vectorial:
  - Programación manual o paralelismo explícito.
  - Vectorización automática o paralelismo implícito.



# PARALELISMO IMPLÍCITO

## ○ Ventajas:

- El programador no se centra en la división de tareas.
- Facilita el diseño de programas paralelos.
- En especial, aplicaciones que realizan operaciones intensivas en matrices

## ○ Inconvenientes:

- reduce el control que el programador tiene sobre la ejecución paralela.
- dificultades en la vectorización del código debidas a las instrucciones de control, dependencia de datos y las indexaciones indirectas.



# PARALELISMO EXPLÍCITO

## ○ Ventajas:

- Aumenta el control del programador.
- Se obtienen programas más óptimos.

## ○ Inconvenientes:

- Actualmente, existen pocos lenguajes vectoriales.
- No existe una normalización aceptada para estos lenguajes.
- Depuración dificultosa.



# EJEMPLO PRÁCTICO: PROGRAMACIÓN MANUAL VS VECTORIZACIÓN AUTOMÁTICA

CPUs	400 × 400			500 × 500			600 × 600		
	SISAL	SR	C	SISAL	SR	C	SISAL	SR	C
1	34.5	60.5	28.2	66.9	118.	56.2	115.0	215.	95.6
2	17.7	30.7	14.6	34.5	60.0	29.3	59.6	114.	52.9
3	12.3	20.8	10.2	23.7	40.8	21.2	40.9	77.8	38.0
4	9.12	16.0	8.73	19.2	31.1	16.7	31.9	60.4	29.7

CPUs	SISAL		SR		C
	<i>osc</i>	<i>fsc</i>	<i>bag</i>	<i>prune</i>	
1	26.7	23.7	25.8	18.2	7.79
2		12.1	15.3	9.10	3.87
3		8.17	13.3	6.07	2.61
4		6.65	22.5	5.64	2.18



# COMPARATIVA ARQUITECTURA ESCALAR Y VECTORIAL

- Usaremos el programa que resuelve:

$$Y=aX+Y$$

- Para la arquitectura escalar usaremos el simulador DLX (14 instrucciones).
- Para la arquitectura vectorial usaremos el simulador DLXV (6 instrucciones).
- En un computador vectorial podemos distinguir dos tiempos en la ejecución de instrucciones:
  - Tiempo de arranque.
  - Velocidad de inicialización.



# COMPARATIVA ARQUITECTURA ESCALAR Y VECTORIAL (II)

## RESULTADOS

- Escalar
  - Ciclos 104
  - 10 saltos condicionales (10,17%)
  - N° de detenciones 44 (42,31)
- Vectorial
  - Ciclos: 31
  - Instrucciones: 21
  - CPI:1,476



# COMPARATIVA ARQUITECTURA ESCALAR Y VECTORIAL (CONCLUSIONES)

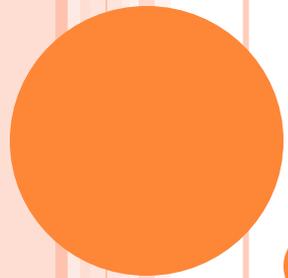
- El rendimiento de DLXV (vectorial) es muy superior (31 ciclos frente a 104).
- En DLX ocurren 44 detenciones.
  - Cada ADD debe esperar por un MULTD
  - Cada SD debe esperar por un ADDD.
- En DLXV no ocurre ninguna.
  - Cada instrucción vectorial opera sobre todos los elementos del vector independientemente.
- DLX necesita 6 operaciones de salto las cuales no afectan a la productividad.



# BIBLIOGRAFÍA

- <http://www.infor.uva.es/~bastida/Arquitecturas%20Avanzadas/Vectoriales.pdf>
  - Introducción general, definiciones básicas, características, tiempo de arranque...
  - <http://atc2.aut.uah.es/~acebron/cap3vect.pdf>
    - Nociones generales, juego de instrucciones, arquitectura DLXV
  - <http://www.angelfire.com/ca6/angie/vectoriales.htm>
    - Información muy general
  - Artículo “A Comparison of Implicit and Explicit Parallel Programming” de Vincent W. Freeh (Universidad de Arizona).
    - Ejemplos de comparación de programación paralela explícita e implícita.
  - Práctica 4 “Procesadores Vectoriales” Universidad de Alcalá.
    - Familiarización con arquitectura DLXV
  - Tema 4 Computadores Vectoriales DIESA.Arquitectura y Tecnología de computadores.
  - Multi-core Programming: Implicit Parallelism. Transparencias de Tuukka Haapasalo
    - Paralelismo explícito e implícito; definiciones, ventajas y desventajas...





**FIN**